Comparison of Knowledge Distillation Methods for Semantic Segmentation

David Courtis

Dec 03, 2024

Section 1. Model Details

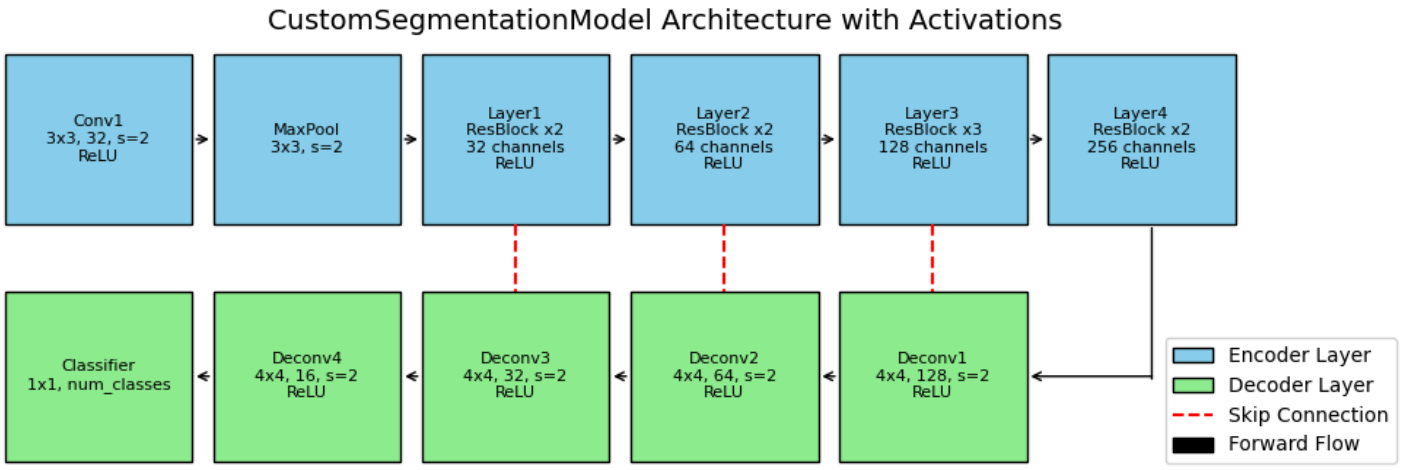

Fig 1. Architectural Diagram of our CustomSegmentationModel

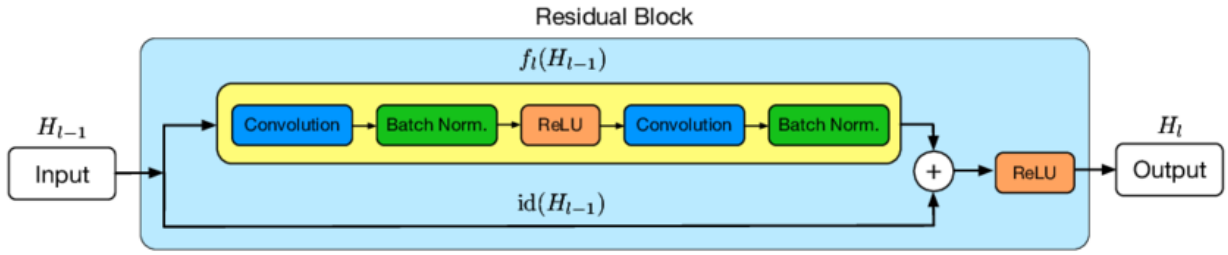

Fig 2. Architectural Diagram of well-known ResBlock

The proposed architecture as shown in **Fig 1**, **Fig 2**, is a lightweight encoder-decoder network inspired by several successful semantic segmentation approaches. The network combines residual learning principles with efficient feature extraction and upsampling pathways. The encoder pathway begins with an initial 3×3 convolution layer that processes RGB input using 32 filters with a stride of 2 and padding of 1[3]. This is followed by batch normalization and 3×3 max pooling with stride 2, which helps reduce spatial dimensions while maintaining important features. The initial downsampling strategy is similar to successful architectures like U-Net but with fewer initial filters to maintain efficiency[2].

The core encoder comprises four sequential blocks of residual layers, progressively increasing the channel depth (32→32→64→128→256) while reducing spatial dimensions. Each residual block contains two 3×3 convolutional layers with batch normalization and ReLU activation, following the original ResNet design principles[1]. The residual connections include 1×1 convolutions when spatial dimensions change, which has been shown to effectively handle feature map matching while minimizing parameters[3].

The network employs skip connections from encoder to decoder, similar to U-Net's architecture, but with residual learning principles. These connections help preserve fine-grained spatial information that might be lost during downsampling[2]. The skip connections are implemented through element-wise addition rather than concatenation, reducing memory requirements while maintaining gradient flow[3].

The decoder pathway consists of four transposed convolution layers (4×4 kernel, stride 2, padding 1) that progressively upsample the feature maps while reducing channel depth (256→128→64→32→16). Each upsampling operation is followed by

feature fusion with the corresponding encoder level through residual connections[4]. This design allows the network to recover spatial details while maintaining computational efficiency.

The final classification layer uses a 1×1 convolution to map the 16-channel feature space to the required number of semantic classes (21 for PASCAL VOC). The relatively shallow depth and careful balance of feature channels make it suitable for real-time applications while maintaining reasonable segmentation accuracy[4]. The network has 3.7m parameters.

Section 2. Knowledge Distillation

Knowledge distillation was utilized to assist with training our custom compact model using a Teacher-Student model, by leveraging the outputs of a pre-trained teacher model. The teacher model used in this implementation was the FCN-ResNet50 model, with weights pre-trained on the VOC Segmentation dataset. This teacher model provided refined feature representations for the student model to mimic. The process was guided by a joint loss function combining supervised loss and distillation loss, ensuring the student learns from both the ground truth and the teacher's predictions. These loss functions are defined below.

- Supervised loss: $L_{supervised}$= H(y, σ(Z_s ; T=1)) where H(,) is the cross entropy loss, y are the ground truth labels, σ(Z, T) is the modified Hinton-style softmax function, Z_s are the student model's predictions, and T is the temperature parameter.
- Distillation loss: $L_{distillation}$= H(σ(Z_t ; T), σ(Z_s ; T)) where Z_t are the teacher model's predictions, and T is the temperature parameter.
- Combined loss: L=αLsupervised+βLdistillation , this loss is back propagated through the network.

During the training loop, the input images are passed through both the student models, and the frozen teacher model. The joint loss is backpropagated through the student model to update its weights using the Adam optimizer. Early stopping monitors validation loss to prevent overfitting, and a ReduceLROnPlateau scheduler also dynamically adjusts the learning rate based on validation performance. With this training architecture, the teacher model guides the student to learn high-level feature representations and relationships between classes, improving segmentation performance. By combining supervised learning with distillation, the smaller student model achieves better generalization compared to relying on ground truth alone.

The knowledge distillation process in our implementation combines both response-based and feature-based distillation approaches to effectively transfer knowledge from the FCN-ResNet50 teacher to our compact student model.

Response-Based Distillation

The response-based distillation focuses on matching the output probability distributions between teacher and student models. For each input image x, we compute:

$$\mathcal{L}_{response} = \mathcal{H}(\sigma(z_t/\tau), \sigma(z_s/\tau))$$

where z_t and z_s are the logits from teacher and student models respectively, τ is the temperature parameter set to 4, and σ is the softmax function. The temperature parameter softens the probability distributions, revealing more information about inter-class relationships.

Feature-Based Distillation

We implement feature-based distillation by aligning intermediate representations across four levels of the network. The feature alignment loss is computed using cosine similarity:

$$\mathcal{L}_{feature} = \frac{1}{N}\sum_{i=1}^{N}(1 - \cos(f_t^i, f_s^i))$$

where f_t^i and f_s^i are the normalized feature maps from the i-th level of teacher and student networks. To handle different spatial dimensions, we use bilinear interpolation to resize teacher features to match student dimensions.

Combined Training Process

Distillation can be performed using one of the above or both. The loss function describe both both combines cross-entropy supervision with both distillation components:

$\mathcal{L}_{total} = \alpha\mathcal{L}_{ce} + \beta(\mathcal{L}_{response} + \mathcal{L}_{feature})$ where $\alpha = 0.7$ and $\beta = 0.3$.

During training, the teacher model's weights remain frozen, input images are processed through both networks, feature maps are extracted at each level, losses are computed and combined, and gradients are backpropagated through the student only. The TeacherWrapper class handles the projection of teacher features to match student dimensions using 1×1 convolutions, ensuring compatible feature comparison despite architectural differences. This dual distillation approach enables the student to learn both high-level semantic information and low-level feature representations from the teacher.

Section 3. Training Hyperparameters and Loss

The training process incorporated several key hyperparameters and training configurations for both the baseline and knowledge distillation approaches.

Base Parameters	
Max epochs	300
Batch size	16
Learning rate	1e-3
Weight decay	1e-4
Input size	224 * 224
# Classes	21
Optimizer	ADAM
Loss Fn	CrossEntropyLoss

Knowledge Distillation Parameters	
Temperature	4
Alpha	0.7
Beta	0.3
Teacher Weights	FCN-ResNet50

Training Resources	
Per Epoch Time	8-9s
GPU	RTX 3090

Optimization Strategy	
lr scheduler	ReduceLROnPlateau
Patience	10
Minimum Delta	0
Factor	0.2
Minimum	1e-7
Early stopping	30
Stopping Metric	IoU

Table 1. Training Base, KD, and Scheduler Hyperparameters. Training Resources.

Data Augmentation	
Random horizontal flipping	50%
Random rotation	±10 degrees
Random affine transformations	Translation: ±0.1, Scale: 0.9-1.1, Shear: ±5 degrees
Color jittering	Brightness: ±0.2, Contrast: ±0.2, Saturation: ±0.2, Hue: ±0.1
Dataset	VOCSegmentation 2012

Table 2. Data augmentation techniques and details.

Loss Plots

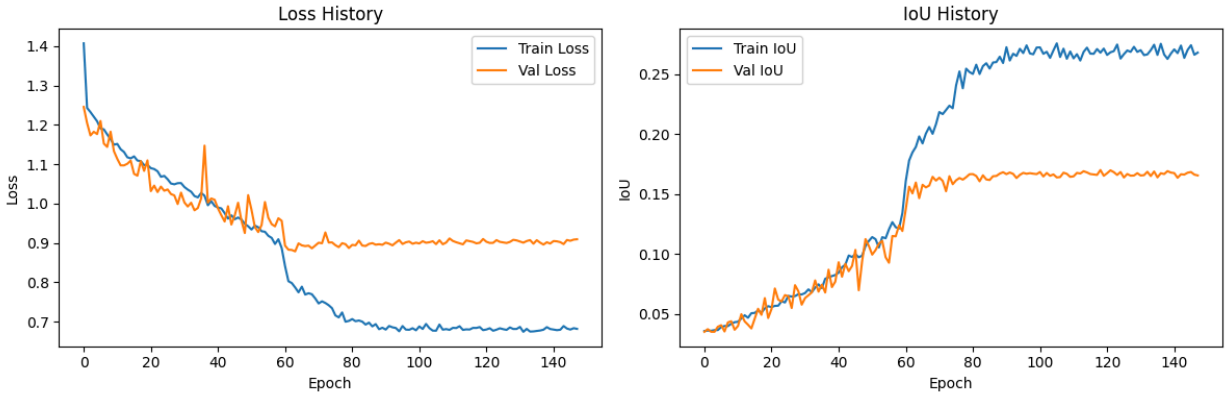

Figure 3. No knowledge distillation applied to the training of the compact model.

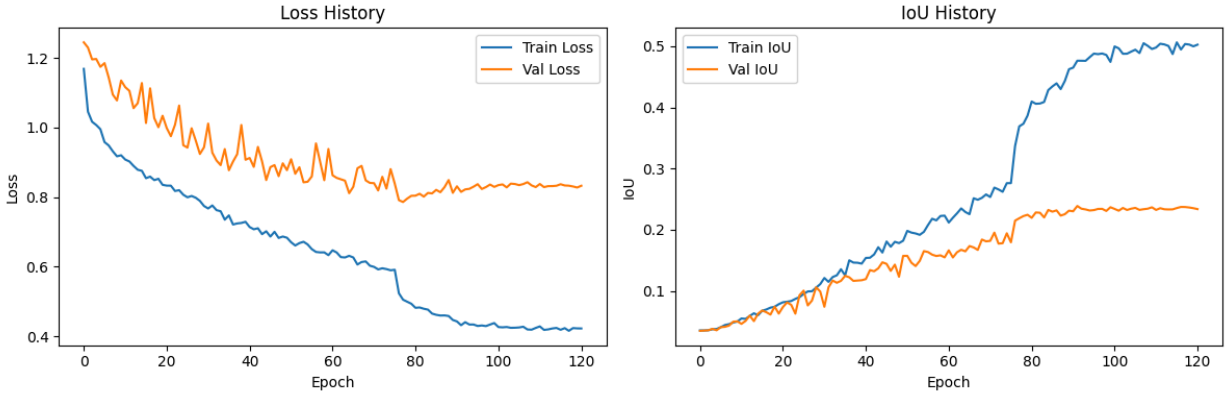

Figure 4. Feature based knowledge distillation applied to the training of the teacher-student model.

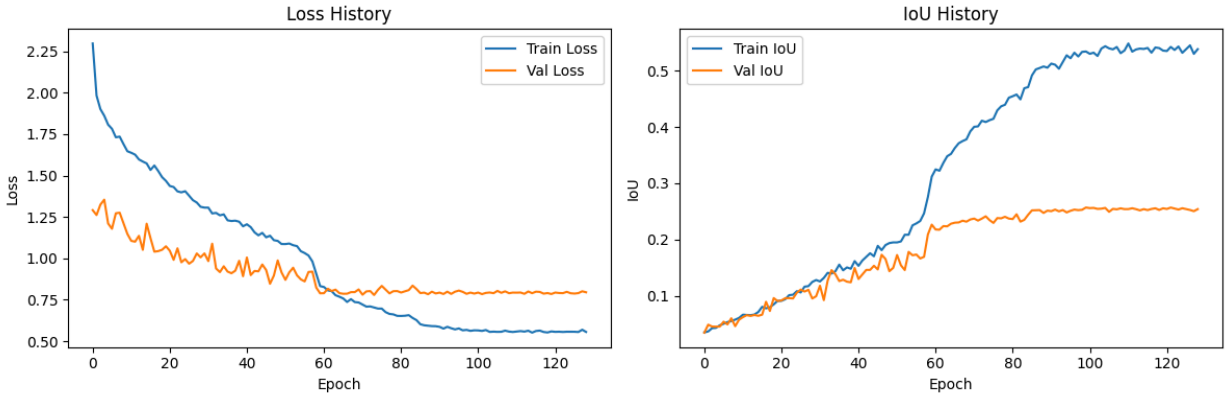

Figure 5. Response based knowledge distillation applied to the training of the teacher-student model.

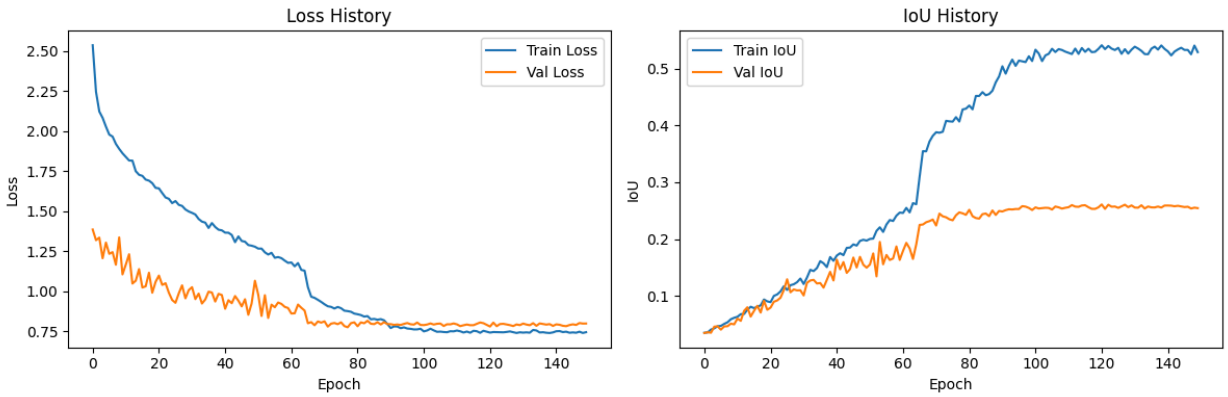

Figure 6. Both Response and Feature based knowledge distillation techniques applied to the training of the teacher-student model.

Section 4. Experiments and Results

Experimental Results:

The experiments were conducted to evaluate the impact of different knowledge distillation (KD) strategies on the performance of a compact segmentation model trained on the VOC 2012 dataset. Training and testing were performed on an NVIDIA RTX 3090 GPU with 24 GB of VRAM. This hardware enabled efficient handling of large batch sizes and reduced training time due to high computational throughput.

The mean Intersection over Union (mIoU) was used as the primary accuracy metric, computed using the torchmetrics library's JaccardIndex implementation. Training employed a batch size of 16 with an initial learning rate of 10−310^{-3}10−3, reduced dynamically using a plateau-based scheduler. The impact of KD methods was assessed based on quantitative results (mIoU, training epochs, total training time) and qualitative visualizations of segmentation masks.

Quantitative Results:

The table below summarizes the results of training with and without knowledge distillation. The experiments evaluated three types of KD strategies:

Knowledge Distillation Type	mIoU	# Training Epochs	Total Training Time
None	0.1699	148	1325s
Feature	0.2389	121	1024s
Response	0.2571	129	1094s
Feature & Response	0.2609	150	1209s

Training without knowledge distillation resulted in the lowest mIoU of 0.1699, indicating poor generalization. Response-based knowledge distillation outperformed feature-based knowledge distillation, achieving a higher mIoU of 0.2571 with relatively stable training. However, combining Feature and Response knowledge distillation methods led to the best performance, achieving the highest mIoU of 0.2609, albeit at the cost of slightly longer training.

Qualitative Results:

To further support the quantitative results, segmentation masks predicted by the model were visualized. Below are examples of successful predictions and then failure cases for each KD method.

Figure 7: Visualization examples of no knowledge distillation model.

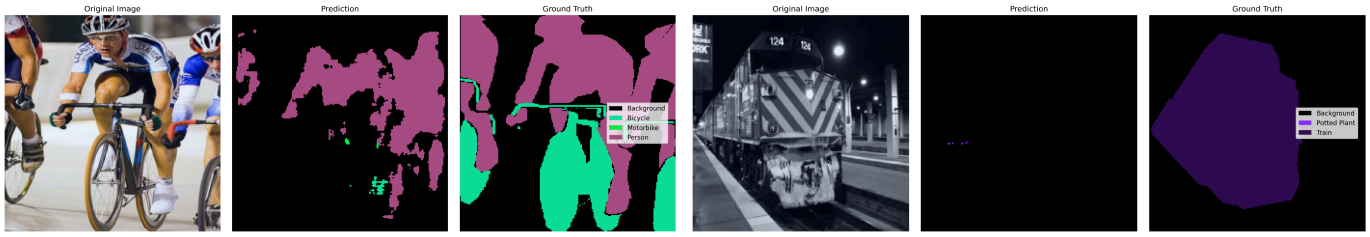

Figure 8: Visualization examples of response-based knowledge distillation model.

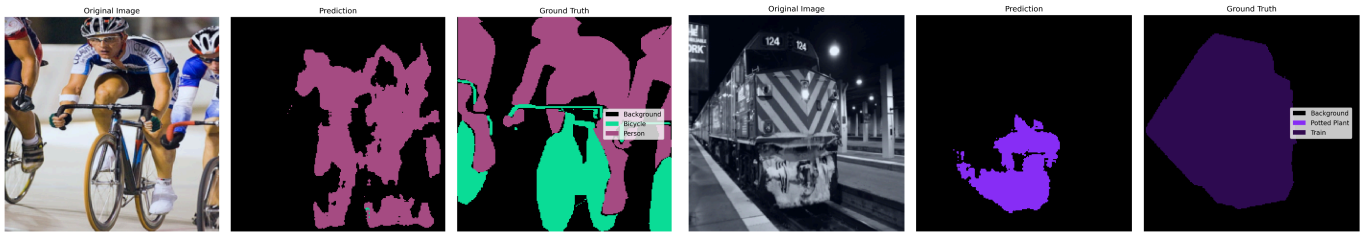

Figure 9: Visualization examples of feature-based knowledge distillation model.

Figure 10: Visualization of Feature & Response based knowledge distillation model.

Knowledge distillation significantly improved performance, with the best results achieved by combining feature and response knowledge distillation. Models trained with knowledge distillation were more stable, with smaller gaps between validation and training losses, indicating better generalization. Response-based knowledge distillation provided a balanced trade-off between mIoU and training time, while combined knowledge distillation delivered the highest accuracy at a slight computational cost. These results demonstrate that incorporating teacher guidance through distillation can effectively enhance compact models for semantic segmentation.

Section 5. Discussion

As expected, the compact model trained without any knowledge distillation techniques produced subpar results in evaluation metrics compared to both the pre-trained ResNet50 segmentation model baseline and the compact model trained with distillation. The limitations in complexity and representational power of the compact model, compared to the much larger ResNet50, made it challenging to generalize well without guidance. Despite this, the compact model achieved reasonable performance considering its significantly reduced parameter count (~2–10 million parameters compared to ResNet50's ~23 million).

The experiments demonstrated that knowledge distillation significantly improved the performance of the compact model. The best results were achieved when combining feature-based and response-based knowledge distillation, which outperformed both individual distillation approaches. This indicates that combining high-level semantic information (response distillation) with low-level feature alignment (feature distillation) enables the student model to better learn from the teacher's representation.

For example, Response-based distillation provided a significant improvement in mIoU, achieving **0.2571** compared to **0.2389** for feature-based distillation alone. This method encouraged the student to replicate the teacher's softened class

probabilities, effectively transferring inter-class relationships. Combined distillation yielded the highest mIoU of **0.2609**, demonstrating that the complementary nature of feature and response guidance can maximize the compact model's learning potential.

Challenges and Solutions

1. **Designing a Lightweight Model**: Designing a model with ~2–10 million parameters that balanced efficiency and accuracy was challenging. Our initial attempt resulted in a 23-million-parameter model that performed poorly. To address this, we researched how ResNet50 was structured and incorporated similar principles, such as residual blocks, into the compact architecture. This redesign achieved better performance while maintaining efficiency.

2. **Effectiveness of Knowledge Distillation**: Knowledge distillation was not immediately effective without data augmentation techniques. Without augmentation, the model tended to overfit to the training set, limiting its ability to generalize. Incorporating techniques such as random flipping, rotation, affine transformations, and color jittering forced the model to focus on learning high-level, invariant features. This complemented the knowledge provided by the teacher model, enabling better generalization.
3. **Hyperparameter Tuning**: Tuning the hyperparameters for knowledge distillation introduced additional complexity to the training process. Parameters such as α, β, and τ required trial and error to optimize alongside the base model's learning rate, weight decay, and scheduler settings. This optimization was time-intensive and required research into recommended values for distillation-specific parameters. Despite these efforts, there remains a possibility that the chosen parameters were suboptimal.
4. **Feature-Based Distillation Implementation**: Feature-based distillation presented an implementation challenge, particularly in projecting the teacher's feature maps to match the student's lower-dimensional representations. To address this, we developed a utility that automatically applied 1×1 convolutions to align the dimensions, enabling effective feature-level comparison and loss computation.

Failure Cases

Analyzing the failure cases revealed insights into the limitations of the compact model:

- **No KD**: The compact model struggled to distinguish fine-grained details or smaller objects, likely due to the lack of guidance from a more complex network. The model frequently produced coarse masks with significant misclassifications.

- **Feature KD**: While feature-based KD improved spatial understanding, it often failed to distinguish semantically similar classes. This suggests that low-level features alone are insufficient for complex segmentation tasks.

- **Response KD**: This approach performed better than feature KD but sometimes misclassified small objects or objects in challenging lighting conditions, indicating that high-level information alone is not enough for robust segmentation.

- **Combined KD**: While this approach produced the best results, it occasionally failed when the teacher model itself struggled with ambiguous or occluded objects, highlighting the dependency on the teacher's limitations.

The failure cases highlight how the compact model's limited capacity impacts its ability to independently learn complex features:

1. **Teacher Dependency**: The compact model depends heavily on the teacher for transferring high-level abstractions, such as semantic boundaries and inter-class relationships. When the teacher's outputs were ambiguous or incorrect, the student model was also likely to fail.
2. **Feature Misalignment**: Even with feature-based KD, the compact model's reduced depth and parameter count meant it lacked the capacity to fully utilize the high-level abstractions distilled from the teacher, especially for complex or edge-case examples.
3. **Inherent Trade-offs**: Each KD method excelled in certain areas but had inherent trade-offs. Feature KD emphasized spatial alignment at the cost of semantic depth, while response KD prioritized semantic relationships but sometimes neglected spatial accuracy.
4. **Limited Capacity**: The compact model's small size inherently restricted its ability to capture intricate details or nuanced distinctions present in higher-level features. This constraint was most evident in cases involving fine-grained segmentation or objects with significant inter-class similarity.

These hypotheses suggest that while KD effectively bridges the gap between the compact model and its teacher, the inherent limitations of the compact model's architecture prevent it from fully replicating the teacher's performance. Further improvements might involve hybrid KD methods or slight increases in model complexity without compromising efficiency.

Analysis

The final compact model with combined distillation achieved an mIoU of **0.2609**, which is a significant improvement over the **0.1699** mIoU achieved without KD. While the gap between the compact model and ResNet50's performance remains substantial, this demonstrates that KD effectively bridges some of the performance disparity. The improvement was achieved with a model that was 8–9 times smaller than ResNet50, validating the effectiveness of knowledge distillation in enabling lightweight models to perform well.

Citations

[1] “Deep residual learning for image recognition,” IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/7780459

[2] O. Ronneberger, P. Fischer, and T. Brox, “U-NET: Convolutional Networks for Biomedical Image Segmentation,” in Lecture notes in computer science, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[3] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” arXiv.org, Nov. 14, 2014. https://arxiv.org/abs/1411.4038

[4] “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/7913730