HS3: Learning with Proper Task Complexity in Hierarchically Supervised Semantic Segmentation

Shubhankar Borse[1]
sborse@qti.qualcomm.com
Hong Cai[1]
hongcai@qti.qualcomm.com
Yizhe Zhang[2]
yizhe.zhang.cs@gmail.com
Fatih Porikli[1]
fporikli@qti.qualcomm.com

[1] Qualcomm AI Research
San Diego, CA, USA
Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

[2] Nanjing University of Science and Technology
Nanjing, China
Work done at Qualcomm AI Research.

Abstract

While deeply supervised networks are common in recent literature, they typically impose the same learning objective on all transitional layers despite their varying representation powers.

In this paper, we propose Hierarchically Supervised Semantic Segmentation (HS3), a training scheme that supervises intermediate layers in a segmentation network to learn meaningful representations by varying task complexity. To enforce a consistent performance vs. complexity trade-off throughout the network, we derive various sets of class clusters to supervise each transitional layer of the network. Furthermore, we devise a fusion framework, HS3-Fuse, to aggregate the hierarchical features generated by these layers. This provides rich semantic contexts and further enhance the final segmentation. Extensive experiments show that our proposed HS3 scheme considerably outperforms deep supervision with no added inference cost. Our proposed HS3-Fuse framework further improves segmentation predictions and achieves state-of-the-art results on two large segmentation benchmarks: NYUD-v2 and Cityscapes.

1 Introduction

Aiming at labeling each pixel to a target category, semantic segmentation is a fundamental task in computer vision for various real-world applications, such as autonomous driving, AR/VR, photography, medical imaging, scene understanding, and real-time surveillance.

Notable advancements in semantic segmentation originated with the end-to-end fully convolutional networks [15]. Researchers have since then looked extensively into various ways to further improve performance by adding different kinds of context to such networks. Some examples are the HRNet [19] branches and hierarchical multi-scale attention [22], which add context based on scale; another similar direction is Object Contextual Representations (OCR) [27], which adds context related to label representations.

In these approaches, deep architectures play a key role, but at the same time, bring challenges to training. For instance, the gradients can vanish as they back-propagate through a

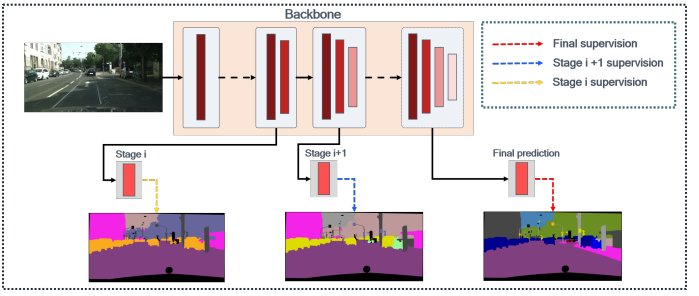

Figure 1: Hierarchical Supervision: Training using our proposed HS3 scheme, where each intermediate supervision uses the right set of classes for its segmentation task, e.g., earlier layers are trained with smaller sets of classes. We show two sample intermediate stages in the figure, with $i \in \{1, ..., N\}$, and the final output stage.

deep network. In addition, the intermediate layers are highly unconstrained and lack prediction capability. In order to address these issues, deep supervision [11, 24] is recently adopted in training, where auxiliary supervisions are imposed on a few selected intermediate layers. To enable intermediate supervision, a separate segmentation head is constructed based on the features of each selected intermediate layer and supervised directly with the original ground truth annotations. In this way, the intermediate layers are tightly regularized by the target task, and more expressive gradients are generated to train the network.

Nevertheless, deep supervision neglects the fact that the intermediate layers have weaker representation powers as compared to the final layer since their features are computed by smaller sub-networks. As such, it can be highly complex for sub-networks to learn to solve the same segmentation problem as the overall network. This prevents the sub-networks from learning meaningful features, and in some cases, can even degrade the overall accuracy (as shown in Section 4). Moreover, the different representation powers of intermediate layers are not taken into account in deep supervision.

In this paper, we propose **Hierarchically Supervised Semantic Segmentation (HS3)**, the goal of which is to find the right learning task for each intermediate layer to be supervised. We attain these segmentation tasks by clustering semantic labels to form a set containing fewer classes, thus less complexity. Specifically, an earlier layer is supervised with a smaller set of classes to match the corresponding sub-network's (the part of the network up to the current layer) learning capacity. We propose a principled approach to determine the number of class clusters using a two-step training process. This approach utilizes the confusion matrices obtained after training a deep supervision baseline to perform automatic hierarchical grouping of classes. Hierarchical supervision is then applied in the second (final) training phase. We show the effectiveness of our method over deep supervision as well as over clustering based on the manual assignment using single-step training.

Now that each intermediate supervised layer can be trained with the suitable grouping of classes, we further propose a framework, HS3-Fuse, to fully utilize the hierarchical features generated by these layers. More specifically, we use lightweight Object Contextual Representation (OCR) modules to process the segmentation features of the supervised intermediate layers. These processed features are then aggregated and fed into the output layer to provide rich hierarchical semantic information and enhance the final segmentation performance.

The contributions of this work are summarized as follows:

- We propose a novel hierarchical supervision scheme, HS3, for training semantic segmentation networks, which allows the supervised intermediate layers to learn with

the right task complexities in terms of the sets of classes. This enhances the feature learning of the intermediate layers without incurring additional inference costs.

- We devise a novel framework, HS3-Fuse, to fully exploit the hierarchical features generated by the intermediate supervised layers. The fused features contain proper and useful hierarchical semantic context and are fed into the output layer to enhance the overall segmentation performance.
- We evaluate our proposed approach on the common benchmarks of Cityscapes, NYUD-v2 and CamVid. The results show that by utilizing HS3, we considerably improve upon the common deep supervision. HS3-Fuse then further improves the accuracy of the segmentation and achieves state-of-the-art performance.

2 Related Work

Semantic Segmentation: The introduction of fully convolutional networks (FCNs) paved the way for significant progress in semantic segmentation [15]. More recent works aim to maximize segmentation accuracy while maintaining a low inference cost, e.g., DeepLab [3], PSPNet [32], and HRNet [24]. Several works then build upon these backbone architectures to incorporate diverse contextual models. The added context could be based on boundaries [1, 21, 28], multi-scale context [13, 22, 26, 27] or relational context [9, 27, 30].

Deep Supervision: Deep supervision was initially proposed to train classification networks [11, 20] and later extended to other tasks, e.g., segmentation [24], depth estimation [6]. These methods, however, assign the same task for all intermediate supervisions, ignoring the weaker learning abilities of sub-networks. Recently, [12] proposes to use intermediate geometric concepts to deeply supervise a key-point estimation network. While the different capacities of intermediate layers are considered, this method is not applicable to segmentation.

Coarse-to-Fine methods: Some recent works apply different coarse-to-fine ideas to improve segmentation, e.g., increasing spatial resolution [8, 14], mask refinement [9, 10, 16]. Our method differs from these as we develop a strategy of class grouping. We use a method of matching task complexities based on sub-network capability, and hence the refinement occurs in an implicit manner. [7] proposes to use different sets of classes for supervision during training. However, its class grouping is manually and specifically designed for the face segmentation problem, and hence not applicable to other segmentation scenarios (e.g., driving, indoors). Furthermore, this grouping is static and cannot adapt to different networks. In contrast, our HS3 derives class grouping in an automated, data-driven manner, which can be applied to any segmentation application and adapt to the learning capacity of any network.

3 Hierarchically Supervised Semantic Segmentation

In this section, we describe the Hierarchically Supervised Semantic Segmentation (HS3) training strategy. The first step involves identifying intermediate or transitional layers in deep networks. We use the approach illustrated by deeply supervised networks [11] to obtain transitional layers, which are demarcated by scale. For instance, the HRNet architecture [12] contains four stages with different scale groupings, which we identify as transitional layers. Note that our method would extend to other segmentation architectures since the identification of transitional layers can also be based on the depth of the layers. Once we identify transitional layers, we train our backbone network by imposing auxiliary supervision through segmentation heads attached to these intermediate layers. Consider a network trained with the HS3 method for N stages. If S is the set of ground truth predictions, we obtain S_i, which

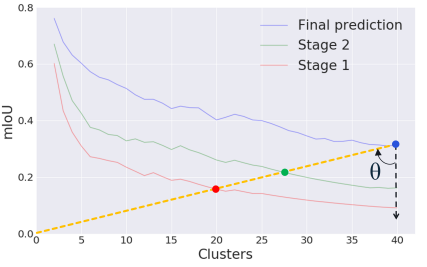

Figure 2: Performance-Complexity Trade-off: We perform the analysis for an HRNetv2-w18-OCR backbone on NYUD-v2. The two intermediate layers are selected based on the scale transitions (more details in Section 4.1). The blue dot indicates the reference trade-off point from the final output. The red and green dots indicate trade-off points for the first and second intermediate supervision stages respectively. The x-axis shows the number of classes after clustering.

is a smaller set of grouped semantic labels for every stage $i, \forall i \in \{1, ..., N\}$. The resulting loss function for HS3 training is given as follows:

$$\mathcal{L}_{\text{total}} = \sum_{i=1}^{N} \gamma_i \mathcal{L}_i^{S_i} + \mathcal{L}_{\text{final}}^{S}, \tag{1}$$

where $\mathcal{L}_i^{S_i}$ is the segmentation loss for the ith intermediate supervision stage, γ_i is the weight of the ith intermediate segmentation loss and $\mathcal{L}_{\text{final}}^{S}$ is the segmentation loss for the final network output. This approach is illustrated in Figure 1.

Our approach differs from deeply supervised networks, for which the set of classes considered for each intermediate supervision is the same as the full set, i.e., $S_i = S$, $\forall i \in \{1, ..., N\}$. However, this scheme imposes the same task complexity on all the intermediate sub-networks, in spite of their weaker and different learning capabilities. Instead, our approach supervises each intermediate layer with an optimal task complexity in terms of the set of semantic classes. We illustrate our approach to finding intermediate semantic sets next.

3.1 Redefining Segmentation Tasks: Learning with the Right Classes

When applying auxiliary supervisions to a deep segmentation network, we allow each supervised intermediate layer to perform a segmentation task that is of the right complexity to it, in terms of the set of classes. We show in this part how to determine this right complexity by analyzing the trade-off between task performance and task complexity.

3.1.1 Segmentation Accuracy vs. Segmentation Complexity

In order to understand the capabilities of the intermediate layers, we first perform a study on the segmentation performance as a function of the task complexity for each of these layers. From the available training data, we reserve a small subset as an *analysis* set and use the rest as a *reduced training* set.[1] First, we train the full segmentation network using vanilla (existing) deep supervision on the *reduced training* set, where all the intermediate supervision stages use the full set of classes. Once the network is trained, we compute the confusion matrix, C_i, for each supervised intermediate layer i, as well as for the final layer, based on the *analysis* set.

[1] For instance, in the case of the Cityscapes dataset, we use 90% of the training data as the *reduced training* set and the remaining 10% as *analysis* set. Note that these sets are always disjoint from the validation/test data.

Next, we study how the segmentation accuracy varies as a function of the number of classes. This indicates the task complexity. Given a target number of classes, we apply spectral clustering [23] to the full set of classes based on an affinity matrix $A_i = (C_i + C_i^T)/2$, which is the symmetric version of the confusion matrix, for each intermediate supervision stage. As the clustering algorithm merges similar classes (e.g., person and rider), we are able to obtain sets of classes with sizes from 2 to $K-1$, where K is the number of classes in the full set. Note that for any two intermediate supervision stages, the sets of classes can be different even when their sizes are the same.

Based on these reduced sets of classes, we re-evaluate the segmentation accuracy for each intermediate stage in terms of mean Intersection-over-Union (mIoU). This analysis reveals the trade-off between segmentation accuracy and segmentation task complexity for each intermediate stage, as well as for the final output layer. Figure 2 shows the trade-off analysis for an HRNetv2-w18-OCR network on NYUD-v2. It can be seen that the accuracy reduces as the task complexity increases (in terms of the number/set of classes) and that an earlier intermediate layer shows weaker capability as compared to a later layer.

3.1.2 Choosing Proper Task Complexity

If the learning task is either too complex or too simple, intermediate layers will not be able to generate useful features to aid the final segmentation. To address this, we utilize the performance-complexity trade-off considering the final output as a reference and enforce the same trade-off across all the intermediate supervision stages. More specifically, we quantify this trade-off using the ratio between the segmentation mIoU and the number of classes. Then, for intermediate supervision stages, we find the trade-off points that match the ratio of the reference point, as highlighted by the green and red dots in Figure 2.

Once the trade-off points are identified, we can readily determine the corresponding numbers of classes, as well as the sets of classes (based on spectral clustering) for the intermediate supervisions. These sets of classes are then used to construct the segmentation losses for the respective auxiliary supervision stages in Eq. 1. We then train the network on the full training set, using the total loss derived from our proposed hierarchical supervisions, which produces the final segmentation model.

In Figure 2, it can be seen that our proposed approach of enforcing consistent performance-complexity trade-offs can be represented by a line through the origin and the reference point (blue dot). We denote the angle between this line and the vertical line through the reference point by θ. By changing θ, one can adjust the trade-offs across the layers. For instance, a larger (smaller) θ places more emphasis on task accuracy (task complexity). In particular, deep supervision corresponds to setting $\theta = 0^\circ$, which requires all the intermediate layers to work on the full segmentation task. As shown in Section 4.4, our proposed approach of enforcing consistent trade-offs achieves a performance very close to the optimum.

3.1.3 Using Other Clustering Methods

Our proposed HS3 framework is general and can be used with any clustering algorithm, as shown in Section 4.4. For instance, instead of running **spectral clustering** on the confusion matrix, one can perform **k-means clustering** based on the features generated by a supervised intermediate layer. This also allows us to analyze the performance-complexity trade-off for each layer, where the merging of the semantic classes is conducted via k-means.

When using spectral clustering or k-means clustering, a two-phase training process is required. It is also possible to train the network only once within our HS3 framework, by

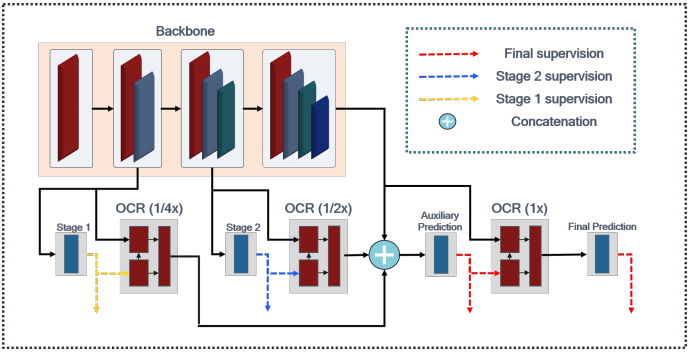

Figure 3: HS3-Fuse: Using the OCR Segmentation Transformer [27] to fuse hierarchical features back into the network.

utilizing a non-data-driven approach to determine the set/number of classes for each intermediate supervision stage. For instance, we can utilize human intuition to manually cluster similar classes and derive reduced sets for the intermediate layers. Another possible way is to set a constant reduction ratio of the number of classes across the layers. For instance, we set the number of classes for each intermediate stage to be $1/2$ of that in the next stage and apply **manual clustering** based on the given number of classes.

3.2 Fusing Hierarchical Features

By utilizing our proposed HS3 approach, the intermediate layers can learn with the right sets of classes, which allows them to generate features of hierarchical semantic contexts at no additional computational cost. We design a fusion framework for aggregating these features to provide richer semantic information to the final segmentation. More specifically, for each intermediate supervision, we feed the segmentation features into an Object Contextual Representation (OCR) block [27], which enhances the features via relational context attention. These enhanced intermediate features are then fused and provided to the final segmentation layer. To reduce computational cost with the task complexity, we set the number of channels in an intermediate OCR block to be $1/2$ of that in the immediate next stage. As we shall see in Section 4, our proposed HS3 and feature fusion allow us to outperform state-of-the-art methods considerably. We illustrate the fusion process in Figure 3 for the case of two intermediate supervision stages. We refer to combining HS3 and feature fusion as **HS3-Fuse**.

4 Experiments

In this section (and in the supplementary file), we present extensive performance evaluations of our proposed approach. We compare HS3 and HS3-Fuse methods with their baseline networks, deep supervision, as well as the latest state of the art. We further conduct ablation studies on our proposed approach.

4.1 Experimental Setup

Datasets: We analyze semantic segmentation performance on two datasets, NYU-Depth-v2 (NYUD-v2) [18] and Cityscapes [5]. We use the original 795 training and 654 testing images for NYUD-v2. We further split the training set into 695 *reduced training* samples and 100 *analysis* samples. For Cityscapes, we use their 2975/500/1525 *train/val/test* splits to report

Network	DS	HS3	mIoU	GMACs (Inference)
HRNetv2-w18-OCR			40.6	22
HRNetv2-w18-OCR	✓		41.2	22
HRNetv2-w18-OCR		✓	**41.7**	22
HRNetv2-w48			47.2	110
HRNetv2-w48	✓		47.0	110
HRNetv2-w48		✓	**47.6**	110

Table 1: On NYUD-v2: Training with the proposed Hierarchical Supervision (HS3) scheme improves performance as compared to various baselines, and also outperforms the Deep Supervision (DS) approach. The improvements come with no added inference cost.

performance. We further split the training set into 2675 *reduced training* samples and 300 *analysis* samples. Models reported on *test* set are trained using *train+val* set.

Metrics: Our primary metric for measuring performance is the mean Intersection-over-Union (mIoU). We also show the mean Pixel Accuracy for our results on NYUD-v2, and the instance IoU (iIoU) for results on Cityscapes. We use GMAC (Multiply-Accumulative Operations in 10^9) to measure computation cost.

Networks: On NYUD-v2, we use HRNetv2-w48 [24], HRNetv2-w18 [24], HRNetv2-w18-OCR [24, 27], and SA-Gate-ResNet-101 [17]. On Cityscapes, we use HRNetv2-w18, HRNetv2-w18-OCR, HRNetv2-w48-OCR, and DeepLab-v3+ [3] with WideResNet-38 (WRN-38) [29] backbone. We apply the intermediate supervisions to layers which transition in scale. For instance, for HRNet backbones, we attach intermediate segmentation heads to the outputs of stages 2 and 3 (and stage 4 generates the final output) [24].

Training: When applying HS3 for training, we select the sets of classes for the intermediate supervision based on our trade-off analysis and spectral clustering in Section 3.1. More details on the training and hyperparameters can be found in supplementary materials.

4.2 Results on NYUD-v2

We report results on the NYUD-v2 validation set. As shown in Table 1, training with our proposed HS3 method consistently improves the performance as compared to deep supervision and the baseline of no intermediate supervision. For HRNet-w48, we observed that deep supervision could even degrade the segmentation performance compared to baseline. Our fusion framework is not used in this comparison. As such, our HS3 approach improves the segmentation accuracy without incurring extra computation cost at inference.

Next, we incorporate the hierarchical predictions into the proposed HS3-Fuse framework. More specifically, we use an SA-Gate-ResNet101 backbone with the proposed fusion unit discussed in Section 3.2. As shown in Table 2, our segmentation performance is 1.2% mIoU more than the SA-Gates baseline. Our HS3-Fuse also achieves better performance when comparing to the latest SOTA on NYUD-v2 using RGB-D inputs, such as InverseForm [1]. Furthermore, we evaluate both single-scale and multi-scale inference schemes (as proposed by [17]) for mIoU and pixel accuracy. Overall, the results indicate that our proposed approach consistently improves segmentation performance in different settings and sets the new SOTA score on NYUD-v2.

4.3 Results on Cityscapes

We provide results on Cityscapes *val* and *test* splits. The results on val with several backbones are summarized in Table 3. We also perform inference by domain adaptation to CamVid dataset [2] using the same weights, as shown in the supplementary file. By using

Network	Backbone	Multi-Scale Inference	mIoU	Pixel-acc
CEN-RefineNet [24]	ResNet-152		51.1	-
SA-Gate [7]	ResNet-101		51.5	76.8
InverseForm [1]	ResNet-101		51.9	77.1
HS3-Fuse (ours)	ResNet-101		**52.2**	**77.4**
Malleable 2.5D [25]	ResNet-101	✓	50.9	76.9
SA-Gate [7]	ResNet-101	✓	52.4	77.9
NANet [31]	ResNet-101	✓	52.3	77.9
CEN-PSPNet [24]	ResNet-152	✓	52.5	77.7
InverseForm [1]	ResNet-101	✓	53.1	78.1
HS3-Fuse (ours)	ResNet-101	✓	**53.5**	**78.3**

Table 2: On NYUD-v2: Comparison with recent state-of-the art RGB-D methods, both with single scale and multi-scale inference. Our proposed HS3-Fuse architecture with a SA-Gates backbone outperforms all other backbones.

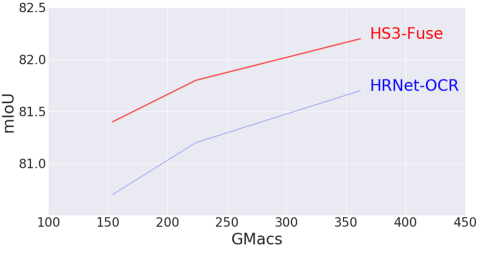

Figure 4: On Cityscapes *val*: Analyzing mIoU v/s GMACs performance with and without the proposed HS-Fuse architecture. We use a backbone HRNetv2-w18-OCR model and tune the OCR parameters to equalize GMAC costs.

our proposed HS3 training scheme, we are able to consistently improve baseline scores as compared to deep supervision. These improvements come with no added inference cost. We further use our proposed HS3-Fuse approach to fully utilize the hierarchical semantic features for the case of HRNetv2-w48-OCR. We achieve an improvement in performance but with additional computational cost during inference. Hence, we also show a lighter version of HS3-Fuse by reducing the number of channels in all OCR modules by a constant factor, such that we match the GMACs required by the baseline model. It can be seen in Table 3 that with the same inference cost, our lighter HS3-Fuse still considerably outperforms the baseline HRNetv2-w18-OCR. Using this technique of scaling OCR channels, we measure performance v/s computational cost at various operating points. As shown in Figure 4, when using the same amount of computation, our proposed approach significantly outperforms the baseline since HS3-Fuse provides richer hierarchical semantic information by fusing our extracted intermediate features.

To evaluate on the test-set, we upload predictions to Cityscapes benchmark server. We use the HS3-Fuse architecture trained using an HRNetv2-w48 [12] with OCR [27] and Hierarchical Multi-scale attention(HMS) [22] model as backbone. As seen in Table 4, We achieve a gain of **0.3** mIoU and **1.7** iIoU over this baseline. We also outperform the previous state-of-the-art model(InverseForm [1]) by a margin of **0.1** mIoU and **0.4** iIoU. Our model ranks top in both categories among published results. We also show visual results comparing our approach to these methods in Figure 5. Details on the predictions obtained from other methods are mentioned in the supplementary file.

Network	Backbone	DS	HS3	mIoU	GMACs
DeepLab-v3+	WRN38	✓		82.8	5.8K
DeepLab-v3+	WRN38		✓	**83.1**	5.8K
HRNetv2-w18	HRNetv2-w18			77.6	76
HRNetv2-w18	HRNetv2-w18	✓		77.7	76
HRNetv2-w18	HRNetv2-w18		✓	**78.1**	76
HRNetv2-w18-OCR	HRNetv2-w18			80.7	154
HS3-Fuse	HRNetv2-w18		✓	**81.8**	224
HS3-Fuse (Lighter)	HRNetv2-w18		✓	81.4	**154**

Table 3: On Cityscapes *val*: Training with the proposed Hierarchical Supervision (HS3) method improves performance compared to various baselines, and also outperforms the Deep Supervision (DS) approach with no added inference cost.

Method	Backbone	mIoU	iIoU
SegFix	HRNet48-OCR	84.5	65.9
Panoptic-DeepLab	Scaled WideResNet	85.1	71.2
Naive Student	WideResNet41	85.2	68.8
Densely-Connected NAS	DCNAS-ASPP	85.3	70.0
Hierarcical Multi-scale attention	HRNet48-OCR-HMS	85.4	70.4
InverseForm	HRNet48-OCR-HMS	85.6	71.4
HS3-Fuse(Ours)	**HRNet48-OCR-HMS**	**85.7**	**71.7**

Table 4: On Cityscapes *test*: Training with the proposed Hierarchical Supervision (HS3) framework achieves state-of-the-art scores among published methods on the live benchmark.

4.4 Ablation Studies

Finding Optimal Number of Clusters: We analyze the segmentation accuracy w.r.t. different performance-complexity trade-offs at the intermediate layers, by varying the parameter of θ. We use an HRNetv2-w18-OCR model on NYUD-v2. Based on the choice of θ, we obtain the numbers of classes K_1 and K_2 for the first and second intermediate stages. As shown in Table 5, the network achieves optimal segmentation mIoU of 41.8% when $\theta = 80°$. By using our proposed approach of enforcing consistent trade-offs across layers from Section 3.1.2, we obtain $\theta = 76°$. This allows us to achieve a near-optimal mIou of 41.7%.

When $\theta = 0°$, we recover the vanilla deep supervision which assigns over-complex tasks to intermediate layers. When $\theta = 90°$, it is required that these intermediate stages achieve the same mIoU as the final output, which results in over-simplified tasks for them. As shown in Table 5, both baselines perform considerably worse as compared to our proposed approach.

Choice of Clustering Methods: We study the effect of using other clustering methods within our hierarchical supervision framework: 1) k-means OCR feature clustering and 2) manual assignment. These are mentioned in Section 3.1.3. We use an HRNetv2-w18 backbone trained using the HS3 scheme and vary our clustering approach. We report our results on the Cityscapes *val* set.

As shown in Table 6, HS3 with any of the clustering methods outperforms deep supervision and the case of no auxiliary supervision. Manual assignment under-performs compared to k-means clustering and spectral clustering, as it is based on human intuition and does not properly align with the sub-networks' capabilities. While k-means clustering performs on par with spectral clustering, it requires class-wise embeddings (e.g., the object representations derived in OCR) at each stage. These representations may not be always available in a given network. In contrast, spectral clustering only requires the confusion matrices.

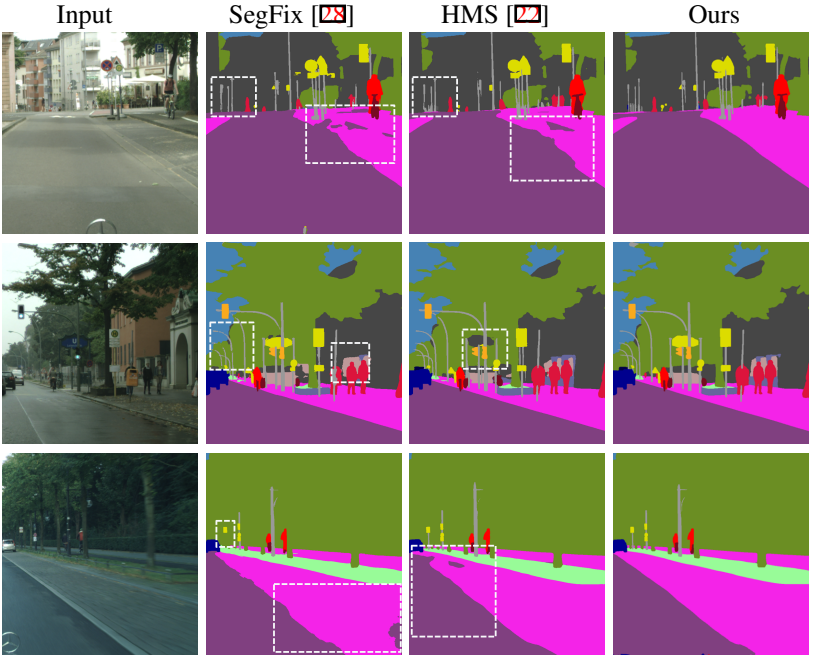

Figure 5: On Cityscapes *test*: Showing visual effect of training an HRNet-OCR-HMS model within the HS3 Fuse framework. Notice the improvement in highlighted regions as compared to previous state-of-the-art works on Cityscapes.

θ	K_1	K_2	mIoU
0°	40	40	41.2
50°	31	34	41.2
75°	20	27	41.7
80°	17	25	**41.8**
85°	10	20	41.4
90°	5	16	41.5

Table 5: On NYUD-v2: Effect of varying θ (i.e., trade-off parameter) in HS3 to train HRNetv2-w18-OCR. Our approach derives a near-optimal $\theta = 76°$ with 41.7% mIoU.

Auxiliary Supervision	Clustering Method	mIoU
None	-	77.6
DS	-	77.7
HS3	manual	77.9
HS3	k-means	**78.1**
HS3	spectral	**78.1**

Table 6: On Cityscapes *val*: Using various clustering methods with HS3 to train HRNetv2-w18. For k-means, OCR modules are used to extract embeddings.

5 Conclusions

In this work, we have presented a training method that supervises the transitional layers of a segmentation network to learn meaningful representations adaptively by varying task complexity. We derived various sets of class clusters to supervise each transitional layer of the network to facilitate this. Furthermore, we devised a fusion framework to leverage additional context offered by our derived hierarchical features. We showed empirically that our proposed training scheme considerably outperforms baselines and also deep supervision with no added inference cost. The proposed fusion architecture offers superior performance on public benchmarks. For future work, we plan to extend our scheme to various tasks, including classification. We're also looking for an acceptable method for single-stage training.

References

[1] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5901–5911, June 2021.

[2] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2): 88–97, 2009.

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[4] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A2-nets: Double attention networks. *arXiv preprint arXiv:1810.11579*, 2018.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[6] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[7] Jiagao Hu, Zhengxing Sun, Yunhan Sun, and Jinlong Shi. Progressive refinement: A method of coarse-to-fine image parsing using stacked network. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 1–6, 2018.

[8] Md Amirul Islam, Shujon Naha, Mrigank Rochan, Neil Bruce, and Yang Wang. Label refinement network for coarse-to-fine semantic segmentation. *arXiv preprint arXiv:1703.00551*, 2017.

[9] Longlong Jing, Yucheng Chen, and Yingli Tian. Coarse-to-fine semantic segmentation from image-level labels. *IEEE Transactions on Image Processing*, 29:225–236, 2019.

[10] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9808, 2020.

[11] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 562–570, 2015.

[12] Chi Li, M Zeeshan Zia, Quoc-Huy Tran, Xiang Yu, Gregory D Hager, and Manmohan Chandraker. Deep supervision with intermediate concepts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1828–1843, 2018.

[13] Di Lin, Dingguo Shen, Siting Shen, Yuanfeng Ji, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Zigzagnet: Fusing top-down and bottom-up context for object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7490–7499, 2019.

[14] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1925–1934, 2017.

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[16] Yadan Luo, Ziwei Wang, Zi Huang, Yang Yang, and Cong Zhao. Coarse-to-fine annotation enrichment for semantic segmentation learning. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 237–246, 2018.

[17] Chen Qian, Hongsheng Li, and Gang Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for RGB-D semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[18] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision*, 2012.

[19] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019.

[20] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[21] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-SCNN: gated shape CNNs for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5229–5238, 2019.

[22] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020.

[23] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

[24] Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. Deep multimodal fusion by channel exchanging. *Advances in Neural Information Processing Systems*, 33, 2020.

[25] Yajie Xing, Jingbo Wang, and Gang Zeng. Malleable 2.5D convolution: Learning receptive fields along the depth-axis for RGB-D scene parsing. In *Proceedings of the European Conference on Computer Vision*, 2020.

[26] Maoke Yang, Kun Yu, Chi Zhang, Zhiwei Li, and Kuiyuan Yang. Denseaspp for semantic segmentation in street scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3684–3692, 2018.

[27] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[28] Yuhui Yuan, Jingyi Xie, Xilin Chen, and Jingdong Wang. Segfix: Model-agnostic boundary refinement for segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 489–506, 2020.

[29] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[30] Fan Zhang, Yanqin Chen, Zhihang Li, Zhibin Hong, Jingtuo Liu, Feifei Ma, Junyu Han, and Errui Ding. Acfnet: Attentional class feature network for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6798–6807, 2019.

[31] Guodong Zhang, Jing-Hao Xue, Pengwei Xie, Sifan Yang, and Guijin Wang. Non-local aggregation for rgb-d semantic segmentation. *IEEE Signal Processing Letters*, 28:658–662, 2021.

[32] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.