Segmenting 2K-Videos at 36.5 FPS with 24.3 GFLOPs: Accurate and Lightweight Realtime Semantic Segmentation Network

Dokwan Oh[1*], Daehyun Ji[1*], Cheolhun Jang[1], Yoonsuk Hyun[1], Hong S. Bae[1], Sungju Hwang[2]

***Abstract*— We propose a fast and lightweight end-to-end convolutional network architecture for real-time segmentation of high resolution videos, NfS-SegNet, that can segement 2K-videos at 36.5 FPS with 24.3 GFLOPS. This speed and computation-efficiency is due to following reasons: 1) The encoder network, NfS-Net, is optimized for speed with simple building blocks without memory-heavy operations such as depthwise convolutions, and outperforms state-of-the-art lightweight CNN architectures such as SqueezeNet [2], MobileNet v1 [3] & v2 [4] and ShuffleNet v1 [5] & v2 [6] on image classification with significantly higher speed. 2) The NfS-SegNet has an asymmetric architecture with deeper encoder and shallow decoder, whose design is based on our empirical finding that the decoder is the main bottleneck in computation with relatively small contribution to the final performance. 3) Our novel uncertainty-aware knowledge distillation method guides the teacher model to focus its knowledge transfer on the most difficult image regions. We validate the performance of NfS-SegNet with the CITYSCAPE [1] benchmark, on which it achieves state-of-the-art performance among lightweight segementation models in terms of both accuracy and speed.**

I. Introduction

Recently, deep neural network architectures for visual recognition have become deeper and wider [7], [8], [9] compared to the deep network models in the past. Thanks to the greatly enhanced learning capacity, these deep and wide networks architectures have achieved remarkable improvements in prediction accuracies, even surpassing human performance in various vision applications (e.g. visual object categorization). Yet, since the focus of research in recent years was mostly on improving the accuracy, the models became heavier and require large memory and computations. However, for many real-world applications of visual recognition such as environment understanding for autonomous vehicles, evaluation speed is a crucial factor since they may require real-time processing of the visual inputs and the models should be often run on devices with limited memory and comptuational power.

Let us now assume a more concrete real-world scenario where we build a scene segmentation system for autonomous vehicles. The first factor we should consider is the amount of computation (operation per second). While conventional deep learning models mostly run on powerful GPU servers, due to security, speed, and network latency issues, remote processing is not a viable option for autonomous driving, and thus the models should run on embedded devices that have significantly limited memory and computation power compared to GPU clusters.

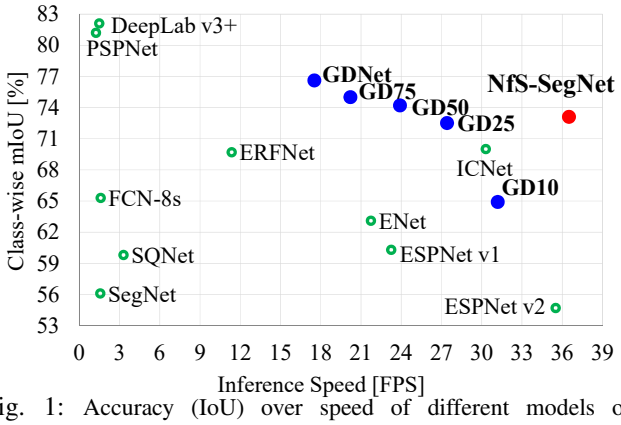

Fig. 1: Accuracy (IoU) over speed of different models on CITYSCAPES [1] leaderboard. NfS-SegNet achieves fastest speed with significantly higher accuracy, compared to baseline real-time semantic segmentation methods.

Second, speed is an important factor in autonomous driving. However, a large portion of the systems on the leaderboard of the CITYSCAPES [1] benchmark focus only on the accuracy with very low frame per second (FPS) and thus cannot be applied to such real-world autonomous driving scenarios. An important caveat here is that reduction of the computation do not always yield increased speed; for instance, if an implementation reduced computations but at the same time significantly increased memory-instensive operations, this implementation may greatly slow down the speed of the model although they may reduce the amount of computation.

Finally, resolution of the video input is another important factor that should be put into consideration when building vision systems for autonomous vehicles. This is because preventive actions such as Adaptive Cruise Control (ACC) or Automatic Emergency Braking (AEB) are effective only when they were done sufficiently early before the accident or collision actually happens. Yet, with low-resolution videos, early detection of objects may be difficult, and thus we need at least Full-HD, or Quad-HD resolution videos as the input. However, this significantly increases the computation cost of the recognition models.

The last challenge is that even with all these difficulties, the model should not compromise its accuracy as it is directly related to safety. To summarize, a recognition model for autonomous driving should process high-resolution videos at high speed, with computing devices that have limited power,

[1] System LSI Division, Samsung Electronics, Korea. {dokwan.oh, derek.ji, c_h.jang, yoonsuk.hyun, hongs.bae}@samsung.com

[*] equal contribution

[2] Korea Advanced Institute of Science and Technology, Korea. sjhwang82@kaist.ac.kr

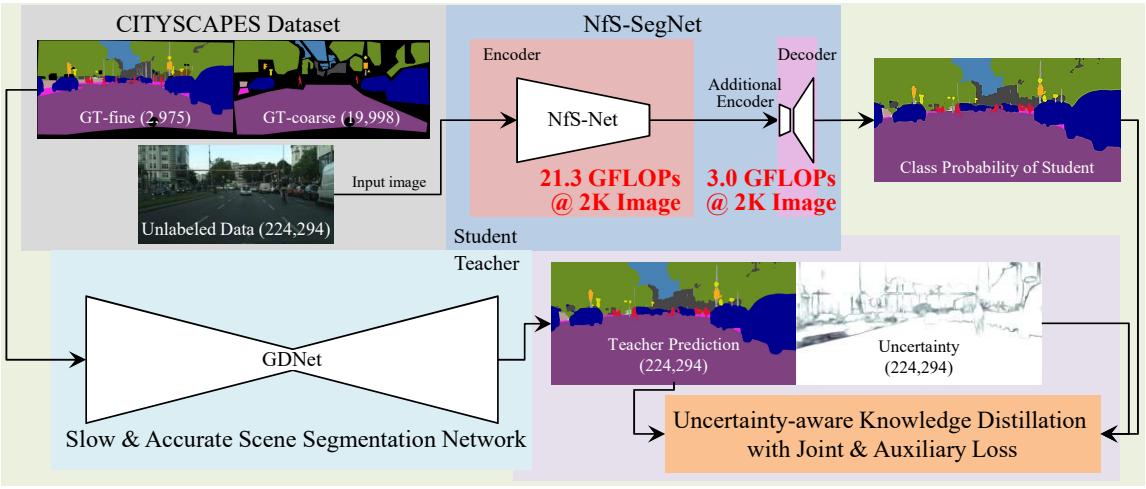

Fig. 2: **System overview:** Our network is composed of a fast encoder and has an asymmetric architecture, where the encoder is heavier than the decoder, and is trained with uncertainty-aware knowledge distillation. The fast encoder network (NfS-Net) and real-time segmentation network (NfS-SegNet) are described in Section III-A and Section III-B. Section IV-A and IV-C introduces our uncertainty-aware knowledge distillation to utilize the knowledge of the larger teacher network (GD-Net) and unlabeled data from CITYSCAPES [1].

without loss of accuracy. How can we then build a model that can meet such strict requirements?

To this end, we propose a fast and lightweight deep network architecture that is trained with novel knowledge distillation method that considers the uncertainty of the model when transferring knowledge for real-time scene segementation. The resulting model, NfS-SegNet, is able to segment 2K-videos at 36.4 FPS with extremely low computation, with significantly higher accuracy compared to existing lightweight segmentation models. We validate NfS-SegNet on the CITYSCAPES [1] challenge, whose results show that our model achieves the fastest speed with accuracy comparable to state-of-the-art systems.

The contribution of this paper is threefold:

- We propose a fast convolutional network architecture, NfS-Net (quoted from Need-for-Speed), that is shallower and achieves significantly faster forward time than state-of-the-art architectures.
- We propose a novel knowledge distillation method called *uncertainty-aware knowledge distillation (U-KD)*, that considers the model uncertainty of the teacher network when transferring knowledge to the student, which significantly outperforms existing knowledge transfer methods.
- We propose an end-to-end trainable architecture for real-time scene segmentation (NfS-SegNet) that uses NfS-Net as an encoder, which obtains the fastest speed on the CITYSCAPES [1] challenge and significantly outperforms state-of-the-art lightweight achitectures.

II. Related Work

a) Fast and lightweight convolutional neural networks: Whilst recent deep convolutional networks have achieved significant improvements on accuracy over the past architectures, their sizes and inference time have been increased proportionally as well. To tackle such increase in computational complexity and inference speed, researchers have explored approaches to reduce the number of parameters in the network while maintaining the accuracy as much as possible. SqueezeNet [2] replaces the majority of 3x3 convolution filters with 1x1 convolution filters, and placed downsampling operations at the upper layer of the network. MobileNet v1 & v2 [3], [4] separate convolution operations into depth-wise and point-wise convolutions, which lead to 8- and 9-fold reduction in computation. Since the main bottleneck in the CNN evaluation is the computation for weights between adjacent layers, ShuffleNet v1 & v2 [5], [6] perform group convolutions and channel shuffling to reduce the computation overhead, which drastically reduces computational complexity. A common limitation of these approaches is that, even with large reduction in computational complexity, the actual runtime does not proportially decrease as well, which is essential in real-time applications. Contrarily, we propose a novel CNN architecture that does not only reduce computational complexity, but also can achieve actual runtime speedups.

b) Knowledge distillation: Knowledge distillation [10] aims to improve the performance of a target network (student) with limited capacity or data, by transferring the knowledge of a pretrained network (teacher), possibly larger or trained with more data. While Hinton et al. [10] proposed to transfer soft labels (network outputs) of the teacher network to the student network, Romeo et al. [11] proposed to perform knowledge transfer at intermediate layers as well, with efficient convolutional regressors. Yim et at. [12] proposed to distill the knowledge as flows between layers, calculated by inner products between features at communicating layers. We target the same problem of compressing the knowledge of a larger network into a smaller one. However, we use uncertainty to let the student network focus knowledge transfer to the regions that are deemed difficult by the teacher, and thus obtain impressive performance improvements over the base KD methods.

c) Uncertainty in deep learning: Gal et al. [20] has shown that deep neural networks trained with dropout regularization is basically a variational approximation of the

posterior of a deep GP, and showed that a dropout-regularized network can output uncertainty by applying dropout at test time. Kendall et al. [19] proposed to capture uncertainty in both the model and the data using MC-dropout and input-dependent variance modeling, and applied it to semantic segmentation and depth estimation. Similarly to Kendall et al., we obtain uncertainty on the teacher network. However, we use uncertainty to measure which features to focus on when transferring knowledge to the student network, which to our knowledge is a novel attempt in utilizing uncertainty in knowledge distillation.

d) Real-Time Semantic Segmentation: Recently, there has been rising interest in building small and efficient neural networks for scene segmentation [15], [51], [16], [17] for real-time segmentation on devices with limited memory and computation power. ENet [15] has a lightweight architecture that is designed from scratch with efficiency in mind, and delivers extremely high speed. ICNet [51] uses the image cascade to speed up the semantic segmentation method. ESPNet [16], [17] is based on a unique convolutional module referred to as efficient spatial pyramid (ESP), which factorizes standard convolution to group point-wise and depth-wise "dilated" separable convolutions instead of expensive point-wise and dilated convolutions. BiSeNet [18] outperformed all previous models in terms of efficiency and accuracy by demonstrating that two types of specific CNN modules (attention refinement and feature fusion module) can perform well, even when the output of network is low resolution. These previous studies on real-time semantic segmentation has focused on decreasing the computational compexity and reducing the network parameters. However, these models are optimized for indirect metrics [6] rather than the actual speed, which can be affected by other factors such as memory swapping cost. We, on the other hand, focus on improving actual inference speed.

III. Need for Speed : End-to-End Realtime Semantic Segmentation Network

To expedite the speed of the deep network for real-time semantic segmentation, we need a fast encoder architecture.

A. NfS-Net: Fast Encoder

In this section, we propose a fast and lightweight convolutional network architecture for the encoder network. We optimize the network for speed by using some of the known techniques. Specifically, we perform all convolutions without bias terms and aggressively downsample in early stage. and promote reuse of the features as much as possible by using the DenseNet [21] structure. We use only four types of layers (convolution, parametric relu, pooling and concatenation). We minimize memory access which is the main bottleneck of the inference. Ma et al. [6] also suggests that excessive group convolution increases memory access cost and network fragmentation reduces degree of parallelism. Fig 3 shows the runtime of NfS-Net at each feature map resolution, against that of well-known fast encoders. We see that NfS-Net does not have any distinctive bottleneck, while MobileNet or ShuffleNet has bottleneck at either the lower layer or upper layer of the network. MobileNet's bottleneck is on use of depthwise convolution that requires heavy memory access, and ShuffleNet is inherently deep in its architecture.

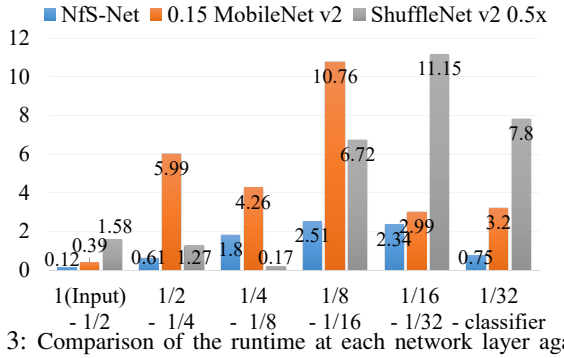

Fig. 3: Comparison of the runtime at each network layer against that of MobileNet v2 and ShffleNet v2 on 2K-input images.

Table I shows that NfS-Net achieves high accuracy despite of its fast forward time, when compared with baseline networks.

Model	Complexity (GFLOPs)	Top-1 err. (%)	FPS
0.15 MobileNet v2 [4]	3.3	55.1	3.0
0.25 MobileNet v1 [3]	3.5	49.4	3.2
0.40 MobileNet v2 [4]	3.7	43.4	1.2
ShuffleNet v1 0.5x [5]	3.2	43.2	19.5
SqueezeNet v1.1 [2]	31.7	42.5	42.6
ShuffleNet v2 0.5x [6]	3.5	39.7	8.6
ShuffleNet v1 1.0x [5]	11.0	35.2	13.0
NfS-Net (proposed)	21.3	**35.0**	**71.4**
ShuffleNet v2 1.0x [6]	12.4	30.6	7.7
1.00 MobileNet v1 [3]	48.4	29.4	0.8
1.00 MobileNet v2 [4]	25.5	28.0	0.6

TABLE I: Comparison to shallow classification networks at 2K input. Although NfS-Net has somewhat higher GFLOPs than the lightest baselines, it is the fastest. Forward times are measured as the average value of 1000 runs in GTX 1080Ti and E5-2620 CPUs using a Caffe implementation, that leverages Cuda 10.0 and Cudnn 7.4.1 libraries.

B. NfS-SegNet: Real-time Semantic Segmentation

For ADAS and autonomous driving, input videos need to have high resolutions to detect objects as early as possible, in order to have sufficient time to react to unexpected events or avoid collisions. To this end, we propose an architecture that can process 2K-videos in real-time. We start by adding a new encoder with a stride at the end of

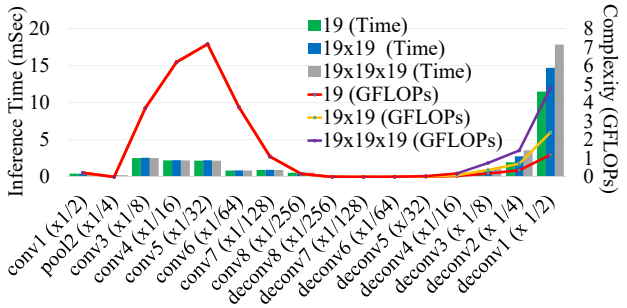

Fig. 4: Time and GFLOPs profile at each network layer for NfS-Seg and its variant. Each legend denotes the filter shape at the last network layer.

NfS-Net, where the input video resolutions are reduced at each step, to obtain a feature map that is as small as 1/256 of the original image. This allows us to lower the amount of computation while dramatically increasing the resolution of the videos. As for the decoder, we made it lightweight, and made it to simply double the result of the decoder. Thus our network architecture is highly asymmetric with most of the computations and parameters allocated for the encoder, with a very shallow decoder. This is in contrast to existing encoder-decoder segmentation architectures, such as UNet [27], which is mostly symmetric. This design choice is based on our empirical findings. First, the encoder has a short runtime even with high computation cost. Figure 4 shows that while the encoder takes up most of the comptuations, the actual runtime is marginal compared to the decoder. Secondly, the increase in the decoder complexity does not contribute much to the performance. Figure 5 shows that increasing the complexity of the decoder results in large reduction in the speed (36.4 FPS $\rightarrow$ 27.5) while obtaining diminishing return on the accuracy (IoU 73.1 $\rightarrow$ 73.4). Based on these findings, we made the encoder relatively heavier while slimming down the decoder.

Our resulting network, NfS-SegNet achieves 36 FPS for 2K video frames in the same condition with Table I. As shown in Table III and Fig 1, NfS-SegNet is significantly faster than the baseline models with remarkably low amount of computation.

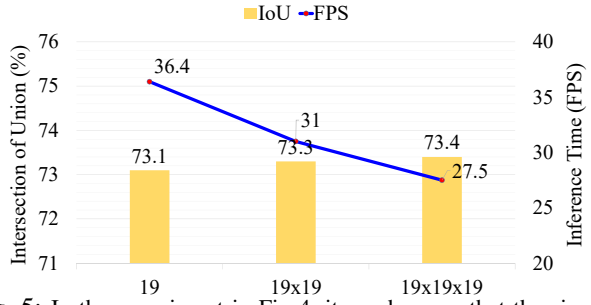

Fig. 5: In the experiment in Fig 4, it can be seen that the simpler the decoder structure, the greater the speed improvement compared to the accuracy reduction.

IV. Improving Model Accuracy with KD

While our lightweight scene segmenation network is fast, it inevitably suffers from performance degeneration when compared to larger networks. To tackle this issue, we use knowledge distillation (KD) [10] with larger teacher networks. For the main experiments, we used a large segmentation network that leverages GoogLeNet-v2 [31] as encoder and DeconvNet [35] as the decoder, which we refer to as GDNet. For verification of the method with various teachers, we also used ENet [15] (lighter than GDNet) and PSPNet [23] (heavier than GDNet) (See Fig 9). The main student model is NfS-SegNet, with GD10, GD25, GD50 and GD75 used to see the KD performance with different compression rates in Section V-B and V-C.

A. Conventional Knowledge Distillation

We first use the conventional knowledge distillation proposed by Hinton et al. [10]. The top left box of Fig . 2 shows

category	# of images	group	# of images
GT-fine	2,975	seen	178,500
GT-coarse	19,998		
unlabeled	155,527		
	45,764	unseen	45,794

TABLE II: Trainable dataset for K.D. except validation and test sequence in CITYSCAPES [1]. *group* is defined for incremental learning scenario of Ch V-B

the example of two types of ground truth provided by the CITYSCAPES. With the labeling quality of GT-Fine, it is expensive to produce labeled data in large quantities, and thus CITYSCAPES only provides $2,975$ training data for GT-Fine subset. For GT-Coarse, however, they provide $19,998$ images since such coarse-level annotations are inexpensive to collect. The teacher prediction of GDNet trained with the union set of GT-Fine and GT-Coarse ($2,975$+$19,998$) is shown at the bottom right box of Fig 2. While the quality of the predicted segmentation is not as accurate as the ground-truth annotations of GT-Fine, it shows significantly better quality compared to ground-truth annotations from GT-Coarse.

With this approach, we can create as much as $224,294$ labeled images. Table III reports the public benchmark performance of the standard KD. Despite the 10% performance improvement (IoU 59.2 $\rightarrow$ 69.2), there is still a large performance gap when compared to the performance of the teacher network. To address this problem, in the following two paragraphs, we introduce improved KD methods, namely joint and auxiliary knowledge distillation and uncertainty-aware knowledge distillation.

B. Joint and Auxiliary Knowledge Distillation

The knowledge distillation approach of Hinton et al. [10] only transfers knowledge as soft labels, which in our case is the pixel-wise class predictions at the final layer of the decoder. However, as done in FitNet [11], we could perform knowledge transfer at intermediate layers to transfer knowledge at different resolutions and abstraction levels, and thus we use a similar scheme but with a different loss. Our auxiliary knowledge distillation loss for the intermediate layers l is defined as follows:

$$\mathcal{L}_{JA}(p_i, \hat{p}_i) = \sum_{i=1}^{N} [\hat{p}_i \log p_i + \alpha \cdot smooth_{L1}(p_i - \hat{p}_i)] \tag{1}$$

in which

$$smooth_{L1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{2}$$

where p_i and $\hat{p}_i$ is the predicted probability logit of the student and teacher network repsectively, α is the hyper-parameter that balances between the cross entropy and the regression term (we use $\alpha = 0.5$), and N is the number of classes.

The main difference of our work from [11] is that we use both the hard and soft transfer by using both the cross entropy loss and the smooth L1 loss, instead of the soft L2 loss used in [11]. We empirically found that this joint training loss made the model to converge faster and converge at higher

test accuracy. Table III shows that the student model trained using knowledge distillation with our proposed auxiliary loss (JA-KD) significantly outperforms the base KD from Hinton et al. [10].

C. Uncertainty-aware Knowledge Distillation

We now propose our novel knowledge distillation approach that considers amount of the uncertainty in the pixel-wise prediction when transferring knowledge from the teacher network to the student network. Based on Kendall et al. [19], uncertainty in prediction can be categorized into 1) aleatoric uncertainty, that comes from inherent ambiguity in data, and 2) epistemic uncertainty, that comes from the model due to lack of data. The aleatoric uncertainty in semantic segementation mostly comes from labeling noise due to annotators' mistakes or variations among annotators. The epistemic uncertainty comes from either confusing classes (e.g. rider and person of Fig 6) or unspecified classes (e.g. backpack of the cyclist in the left image of Fig 6). For more examples, see Fig 6; darker pixels shows pixelwise predictions with high uncertainty.

Since we are mostly interested in distilling the knowledge of the teacher network, we use the epistemic uncertainty for our uncertainty-aware knowledge distillation. Epistemic uncertainty can be obtained by dropout variational inference [20], where we obtain the predictive distribution for each test sample by applying random dropout K-times and aggregating the results. After obtaining the pixelwise uncertainty, we use it as an attention mask to guide which area to focus on, when performing knowledge transfer. The proposed pixelwise loss is defined as follows:

$$\mathcal{L}_U(p_i, \bar{p}_i^{mc}, \bar{u}_i^{mc}) = \sum_{i=1}^{N} \bar{u}_i^{mc} \cdot [\bar{p}_i^{mc} \log p_i + \alpha \cdot smooth_{L1}(p_i - \bar{p}_i^{mc})] \quad (3)$$

where $\bar{p}_i^{mc}$ is mean of the predictive distribution from the teacher model and $\bar{u}_i^{mc}$ is the binarized uncertainty for the i_{th} pixel obtained by median-thresholding on the pixelwise variance, where both the mean and the variance is obtained using MC-dropout with 5 minibatch. The 1% distortion is applied at six layers, which have the most compressed features with low resolution. This formulation may seem counterintuitive, since it guides the knowledge transfer to mostly happen on those regions where the teacher is most uncertain about. However, since the two networks deal with the same input, and those regions with high uncertainty could be also confusing to the student network as well, it is actually beneficial for the teacher to focus on uncertain regions. Fig 6 shows that the uncertain regions are mostly object boundaries which are crucial in achieving high accuracy for semantic segmentation, while the certain regions are mostly inside of the objects that are easy to predict.

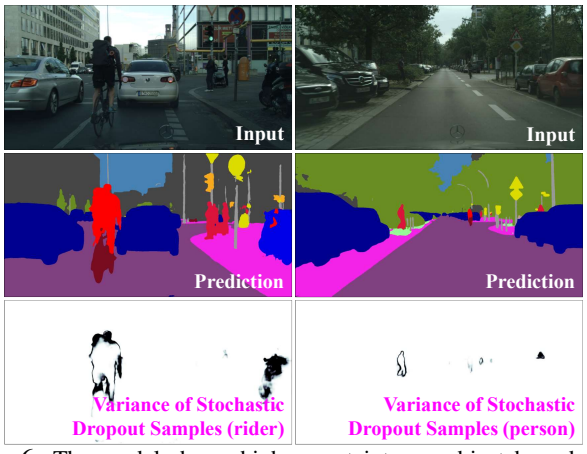

Fig. 6: The model shows high uncertainty on object boundaries and on regions that are difficult to classify, and our U-KD focus on those challenging image regions to perform knowledge transfer.

Model	FPS	GFLOPs	Performance			
			class		category	
			IoU	iIoU	IoU	iIoU
PSPNet [23]	1.3	1093	82.1	62.4	92.0	81.9
Deeplab v3+[24]	1.5	-	81.2	59.6	91.2	79.2
SegNet [19]	1.6	1530	56.1	34.2	79.8	66.4
ERFNet [25]	11.4	-	69.7	44.1	87.3	72.7
ENet [15]	21.7	37.3	63.1	34.1	83.6	63.1
ESPNet v1 [16]	23.3	-	60.3	31.8	82.2	63.1
ICNet [51]	30.3	-	70.0	-	-	-
ESPNet v2 [17]	35.4	-	54.7	28.0	78.7	59.5
GDNet (ensemble)	3.0	141.6	75.7	49.1	88.8	72.9
NfS-SegNet (GT)	**36.4**	**24.3**	59.2	28.7	80.8	59.3
NfS-SegNet (KD)			69.2	40.3	85.8	67.5
NfS-SegNet (JA-KD)			71.0	41.6	86.6	69.4
NfS-SegNet (U-KD)			**73.1**	**44.4**	**87.5**	**70.1**

TABLE III: CITYSCAPES benchmark leaderboard. GT: using only given ground truth, KD: (conventional) knowledge distillation, JA-KD: KD w/ proposed joint & auxiliary loss, U-KD: JA-KD w/ uncertainty-aware loss. The performance of NfS-SegNet is very close to GDNet through the accumulation of the proposed methods, and the final result is better than other real-time segmentations

V. Experimental Results

Now we validate our fast, lightweight semantic segmentation network, NfS-SegNet on real-time video segmentation of first-person dashcam videos, as well as perform experiments to analyze each of its part (fast CNN encoder, uncertainty-aware knowledge distillation).

A. CITYSCAPES benchmark

We report the performance of NfS-SegNet, on CITYSCAPES [1] Pixel-wise Segmentation Benchmark challenge. Since on the leaderboard for runtime speed, there are many teams that inaccurately report the forward time by excluding the time for I/O, and not reporting the actual

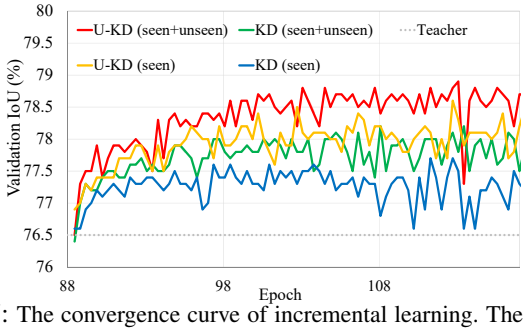

Fig. 7: The convergence curve of incremental learning. The uncertainty derives useful information from unseen data.

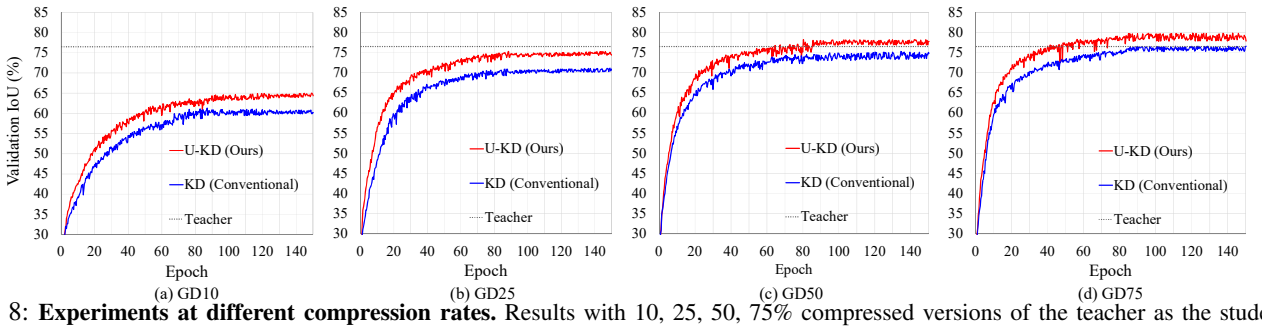

Fig. 8: **Experiments at different compression rates.** Results with 10, 25, 50, 75% compressed versions of the teacher as the student.

inference time, we reproduced the models and report the actual inference speed of each model. As shown in Fig 1 and Table III, our NfS-SegNet achives the fastest speed and smallest amount of computation among the compared models, while also achieving very high accuracy.

B. Incremental Learning with U-KD

Assume that we have an autonomous vehicle running in a city or environment that was never seen during training; then it would be good for the autonomous vehicle to continuously learn from the incoming video inputs. Thus we simulate this on-device learning scenario by experimenting with our U-KD in an incremental learning scenario. We first divide the training data into the seen and unseen groups (See Table II). **seen** is the domain that consists of video frames collected from the same city that the training data was collected.The **unseen** is a set of video frames collected from cities other than ones that were used in training. In Fig 7, the yellow and blue curve represents the accuracy improvements on the already seen domain, and the red and green curves include situations that have not yet been experienced. The results show that uncertainty-aware K.D enhance the performance on its own by concentrating on the patterns that have not been shown during the original training stage. The proposed uncertainty has high value on the out-of-distribution visual patterns, on which our uncertainty-aware knowledge distillation allocate higher weights to consider learning features on those regions more. Fig 7 shows that our U-KD model significantly outperforms base KD, both on seen and on the entire domain. Specifically, U-KD applied to seen domain performed similarly to KD applied to both seen and unseen domains, although it uses much less data. The red curve of Fig 7 achieved an additional improvement of 1.1% over the converged model in the test submit of CITYSCAPES.

C. Additional Experiments

Effect of compression rate on the performance of knowledge distillation We have created new GDNet variants by reducing the number of channels of all convolution layers to 75%, 50%, 25%, and 10% of the full network. We performed knowledge distillation experiments with the full GDNet as a teacher and the network with reduced sizes as students. Fig 8 shows the results of this experiment. We observe that the models compressed with knowledge distillation perform quite well, achieving almost similar or even better performance when using as much as 25%-50% of the full network, and that our U-KD achieves significantly higher accuracy compared to the one trained with base knowledge distillation at any compression rates.

Using different teacher network architectures To show that U-KD can be used with different types of teacher network architectures, we experiment with two different teacher network architectures: 1) PSPNet [23], which is heavier than GDNet with higher performance, and ENet [15] that is lighter than GDNet but achives lower performance. Please refer to Fig 1 for the accuracy and the speed of the two base networks. Fig 9 shows the segmentation performance of our NfS-SegNet using the two network architectures as teachers. We observe that with PSPNet as teachers, NfS-SegNet does not perform as well since the target network is significantly smaller (0.022 times of the teacher network). However, U-KD significantly outperforms base KD. When using ENet (37.3 GFLOPS) as the teacher, the student NfS-SegNet (24.3 GFLOPS) achieves higher accuracy over the teacher network, while requiring significantly less computation.

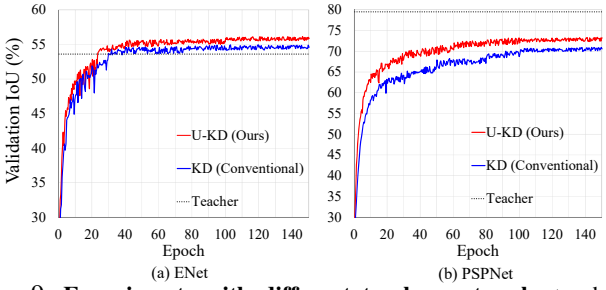

Fig. 9: **Experiments with different teacher networks** teacher: ENet and PSPNet, student: NfS-SegNet. With ENet as teachers, lighter students outperform the heavier teachers.

VI. Conclusion

We proposed a fast and lightweight end-to-end CNN architecture for real-time scene segmentation of high-resolution videos. Our model, NfS-SegNet, is composed of a very fast encoder (NfS-Net), and is designed asymmetrically to allocate the parameters and computation more on the encoder that can be computed fast, with less focus on the decoder which is the bottleneck in inference. We further proposed a novel uncertainty-aware knowledge distillation method, that focuses more on the difficult part of the image when distilling knowledge of the teacher network, and significantly improved the accuracy of our network. We validate our method on CITYSCAPES benchmark, on which it outperforms all other lightweight real-time semantic segmentation models in both the accuracy and the speed.

REFERENCES

[1] Cityscapes Benchmark : https://www.cityscapes-dataset.com/benchmarks/

[2] F.N. Iandola, S. Han, M.W. Moskewicz, K. Ashraf, W.J. Dally, and K. Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 0.5 MB model size. In *ICLR* 2017.

[3] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko W. Wang, T. Weyand, M. Andreetto, and H. Adam. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861v1*, 2017.

[4] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv preprint arXiv:1801.04381*, 2018

[5] X. Zhang, X. Zhou, M. Lin, and J. Sun. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] N. Ma, X. Zhang, H. Zheng and J. Sun. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design, In *The European Conference on Computer Vision (ECCV)*, 2018.

[7] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, In *ICLR*, 2015.

[8] C. Szegedy, S. Ioffe and V. Vanhoucke. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, In *arXiv preprint arXiv:1602:07261*, 2016.

[9] K. He, X. Zhang, S. Ren and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. In *NIPS workshop*, 2014.

[11] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta and Y. Bengio. Fitnets: Hints For Thin Deep Nets, In *ICLR*, 2015.

[12] J. Yim, D. Joo, J. Bae and J. Kim. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning, In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2017.

[13] J. Long, E. Shelhamer and T. Darrell. Fully Convolutional Networks for Semantic Segmentation, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] A. Krizhevsky, V. Nair and G. Hinton. CIFAR-10 (Canadian Institute for Advanced Research) : http://www.cs.toronto.edu/ kriz/cifar.html

[15] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[16] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation. In *ECCV*, 2018.

[17] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu and N. Sang. BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation, In *ECCV*, 2018.

[19] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. In *BMVC*, 2017.

[20] Y. Gal and Z. Ghahramani, Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *ICML*, 2016

[21] G. Huang, Z. Liu, L. Maaten, and K.Q. Weinberger. Densely Connected Convolutional Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] L-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*, 2018.

[25] E. Romera, J.M. Alvarez, L.M. Bergasa, and R. Arroyo. Efficient ConvNet for Real-time Semantic Segmentation, In *IV*, 2017.

[26] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang. Is the deconvolution layer the same as a convolutional layer? *arXiv preprint arXiv:1609.07009*, 2016

[27] O. Ronneberger, P. Fischer and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI* ,2015.

[28] H. Chang, E. Miller, A. McCallum. Active Bias: Training More Accurate Neural Networks by Emphasizing High Variance Samples. In *NIPS*, 2017

[29] J. Shen, N. Vesdapunt, V.N. Boddeti, and K.M. Kitani. In Teacher We Trust: Learning Compressed Models for Pedestrian Detection. *arXiv preprint arXiv:1612.00478*, 2016.

[30] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

[32] P. Zhao and T. Zhang. Stochastic optimization with importance sampling.*arXiv preprint arXiv :1412.2753*, 2014.

[33] D. Meng, Q. Zhao, and L. Jiang. What objective does self-paced learning indeed optimize? *arXiv preprint arXiv:1511.06049*, 2015.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna. Rethinking the Inception Architecture for Computer Vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] H. Noh, S. Hong, B. Han. Learning Deconvolution Network for Semantic Segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[36] Y. Mu, W. Liu, X. Liu, and W. Fan. Stochastic gradient made stable: A manifold propagation approach for large-scale optimization. *IEEE Transactions on Knowledge and Data Engineering*, 2016.

[37] C. G. Northcutt, T. Wu, and I. L. Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. *arXiv preprint arXiv:1705.01936*, 2017.

[38] T. Pi, X. Li, Z. Zhang, D. Meng, F. Wu, J. Xiao, and Y. Zhuang. Self-paced boost learning for classification. *In IJCAI*, 2016.

[39] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015

[40] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[41] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick,S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[42] J. Jin, A. Dundar, and E. Culurciello. Flattened convolutional neural networks for feedforward acceleration. *arXiv preprint arXiv:1412.5474*, 2014

[43] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization, In First Workshop on Fine-Grained Visual Categorization. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*,2011.

[44] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. *arXiv preprint arXiv:1511.06789*, 2015.

[45] M. Wang, B. Liu, and H. Foroosh. Design of efficient convolutional layers using single intra-channel convolution, topological subdivisioning and spatial "bottleneck" structure. *arXiv preprint arXiv:1608.04337*, 2016.

[46] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *NIPS*, 2016.

[47] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng. Quantized convolutional neural networks for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[48] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks.*arxiv preprint arXiv:1611.05431*, 2016.

[49] X. He, R. S. Zemel, and M. Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[50] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, 2009.

[51] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia. ICNet for Real-Time Semantic Segmentation on High-Resolution Images. *CoRR abs/1704.08545*, 2017.

[52] V. Lempitsky, A. Vedaldi, and A. Zisserman. Pylon model for semantic segmentation. In *NIPS*, 2011.

[53] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov. Fast approximate. Energy minimization with label costs. International Journal of Computer Vision, 96, 1–27 (2012).

[54] J. M. Gonfaus, X. Boix, J. Van de Weijer, A. D. Bagdanov, J. Serrat, and J. Gonzalez. Harmony potentials for joint classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[55] P. Kohli, L. Ladicky, P. Torr. Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision, 82, 302–324 (2009).

[56] L.-C. Chen, G. Papandreou, and A. Yuille. Learning a dictionary of shape epitomes with applications to image labeling. In *ICCV*, 2013.

[57] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[58] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams,J. Winn, and A. Zisserma. The pascal visual object classes challenge a retrospective. International Journal of Computer Vision, 111, 98–136 (2015).

[59] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[60] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. "Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[61] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[62] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[63] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8):1915 - 1929*, 2005

[64] G. Lin, C. Shen, I. Reid et al. Efficient piecewise training of deep structured mod. *arXiv preprint arXiv:1504.01013*, 2015.

[65] T.-Y. Lin et al., Microsoft COCO: Common objects in context, In *ECCV*, 2014.

[66] R. Vemulapalli, O. Tuzel, M.-Y. Liu, and R. Chellappa. Gaussian conditional random field network for semantic segmentation, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[67] W. Shi, J. Caballero, L. Theis, F. Huszar, A. Aitken, C. Ledig, and Z. Wang. Is the deconvolution layer the same as a convolutional layer? *arXiv preprint rXiv:1609.07009 [cs.CV]*, 2016.

[68] X. Li, Y. Zhou, Z. Pan, and J. Feng. Partial Order Pruning: for Best Speed/Accuracy Trade-off in Neural Architecture Search, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[69] P. Chao, C. Kao, Y. Ruan, C. Huang, and Y. Lin. HarDNet: A Low Memory Traffic Network, In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[70] Y. Gal. Uncertainty in Deep Learning, *University of Cambridge (PhD Thesis)*, 2016.

[71] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. Le. MnasNet: Platform-Aware Neural Architecture Search for Mobile, In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.