Semantic Segmentation Datasets for Resource Constrained Training

Ashutosh Mishra[1⋆], Sudhir Kumar[1,2⋆], Tarun Kalluri[1,3⋆], Girish Varma[1], Anbumani Subramaian[4], Manmohan Chandraker[3], and CV Jawahar[1]

[1] IIIT Hyderabad
[2] University at Buffalo, State University of New York
[3] University of California, San Diego
[4] Intel Bangalore

Abstract. Several large scale datasets, coupled with advances in deep neural network architectures have been greatly successful in pushing the boundaries of performance in semantic segmentation in recent years. However, the scale and magnitude of such datasets prohibits ubiquitous use and widespread adoption of such models, especially in settings with serious hardware and software resource constraints. Through this work, we propose two simple variants of the recently proposed IDD dataset, namely *IDD-mini* and *IDD-lite*, for scene understanding in unstructured environments. Our main objective is to enable research and benchmarking in training segmentation models. We believe that this will enable quick prototyping useful in applications like optimum parameter and architecture search, and encourage deployment on low resource hardware such as Raspberry Pi. We show qualitatively and quantitatively that with only 1 hour of training on 4GB GPU memory, we can achieve satisfactory semantic segmentation performance on the proposed datasets.

Keywords: Semantic Segmentation, Neural Architecture Search

1 Introduction and Related Work

Semantic segmentation is the task of assigning pixel level semantic labels to images, with potential applications in fields such as autonomous driving [5,16] and scene understanding. Many approaches have been proposed to tackle this task based on modern deep neural networks [18,12,4,14]. Majority of the proposed approaches use encoder-decoder networks that aggregate spatial information across various resolutions for pixel level labeling of images. For example, [12] proposes an end-to-end trainable network for semantic segmentation by replacing the fully connected layers of pretrained AlexNet [8] with fully convolutional layers. Segmentation architectures based on dilated convolutions [17] for real time performance have also been proposed in [18,14]. However most of these approaches come with huge overhead in training time and inference time since it

⋆ equal contribution.

requires multi-GPU training with very high GPU memory requirements. This poses multiple challenges for widespread use of semantic segmentation datasets and architectures, resulting in huge roadblocks for research and development of such real time systems, especially in developing regions of the world with resource constraints. We believe that there are multiple challenges posed by these current approaches. Firstly, compared to image classification tasks on

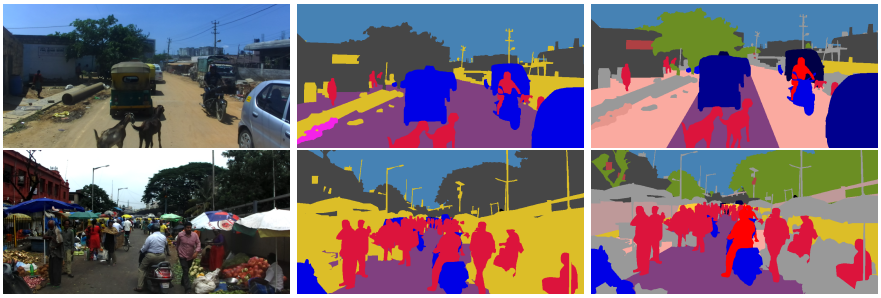

Fig. 1: Sample images with ground truth from IDD-lite, IDD-mini with second and third column representing 7 and 16 labels *(Best viewed when zoomed)*.

datasets like MNIST [9] or CIFAR [8], semantic segmentation is limited in its scope for ubiquitous adoption which essentially rules out the introduction of any such project as part of a curriculum. Although large scale datasets for training the semantic segmentation models such as KITTI [6], CamVid [1] or recently introduced Cityscapes [5] and India Driving Dataset [16](IDD) which provide finely annotated images to train semantic segmentation models with a focus on autonomous navigation exist, the scale and the size of these datasets limit their widespread use, particularly in scenarios where computation resources are limited.

Secondly, navigating through the hyperparameter space for coming up with the most optimum configuration and architecture for semantic segmentation is a demanding task due to huge training costs involved with deep neural networks. In the context of classification, previous works [11] perform architectural search on CIFAR dataset to show that the best performance also applies on larger scale datasets like ImageNet. Several works also use reinforcement learning [2,20], evolutionary algorithm [11] etc. for this purpose. However, there have been fewer works [10,3] to conduct architectural search on dense segmentation task due to resource intensiveness of the task. Hence smaller datasets that enables quick prototyping for hyper parameter search, and help in replicating the results on larger datasets is essential. This would bring down the cost of training, and would aid in improving the overall performance.

Finally, there is a need to drive the research in vision community towards achieving state of the art results for various tasks using only limited labeled data. Such a research direction would have huge impact, more so on semantic segmentation tasks that requires huge annotation of pixel level semantic labels. To

Dataset	Average Resolution	#Annotated Pixels[10^6]	#Train Images	#Val Images	Disk Space (in GB)	Label Size
IDD [16]	968×1678	11811	6993	981	18	26
Cityscapes [5]	1024×2048	9430	2975	500	12	20
IDD-mini	512×720	535	1794	253	4	16
IDD-lite	227×320	39.75	673	110	<1.5	7

Table 1: Comparison of state of the art datasets against proposed IDD-mini and IDD-lite.

address these challenges, we come up with two variants of the recently proposed India Driving Dataset (IDD) [16], namely *IDD-mini* and *IDD-lite*, as shown in Figure 1, which are aimed at improving the state of semantic segmentation for autonomous driving in developing regions. We believe that having these datasets would help alleviate the challenges discussed above in resource constrained settings. Resource constraint can mean lack of availability of high end GPUs, limited time access to GPU resources or lack of infrastructure to store large scale datasets. The scenes and labels presented in our dataset are very different from those available in semantic segmentation datasets such as Cityscapes [5], KITTI [6] or CamVid [1]. Moreover, by developing such standardized small scale datasets, we wish to coalesce the efforts of the research community towards developing algorithms that need only few labels to match state of the art performance.

In summary, our contributions can be stated as follows.

- We provide IDD-mini and IDD-lite, which are subsampled version of IDD with very similar label statistics and smaller number of labels (See section 2).
- We show that models trained only for an hour on a single 4GB GPU still achieve reasonable prediction accuracies, making it possible to include them as part of short courses, workshops and labs in universities and other training centers (See section 4).
- We establish that the accuracy of various models trained on our datasets correlates well with the accuracy on large scale datasets especially in cross-domain setting. This allows for fast prototyping and architectural search for semantic segmentation algorithms (See section 4).
- We deploy models trained using our datasets on Raspberry Pi and report the accuracy and runtime, giving a standardized measurement of the performance characteristics on the device (See section 4).

2 Dataset

We designed the two variants of the datasets with an aim to reduce the overall hardware footprint for storing and processing, keeping intact the diversity and variety from the original IDD dataset. In this section, we present the procedure used to come up with the train-val splits for IDD-mini and IDD-lite. We also provide statistical properties of the proposed datasets, and compare it with the original IDD dataset, along with another state of the art driving dataset, Cityscapes [5].

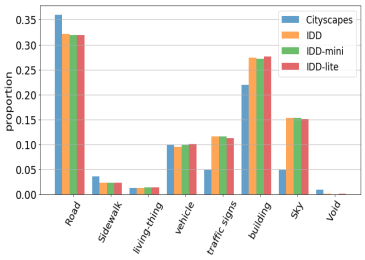

Fig. 2: Proportion of labels in total dataset for IDD, IDD-mini and IDD-lite *(Best viewed when zoomed).*

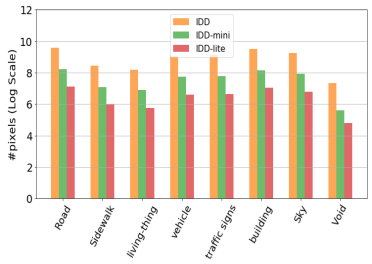

Fig. 3: Absolute value of annotated pixels (in powers of 10) for categories in IDD, IDD-mini and IDD-lite *(Best viewed when zoomed).*

Dataset Specifications

IDD-mini The motivation behind designing IDD-mini is to have a small scale segmentation dataset that is useful for training image segmentation models on low resource hardware. The full IDD dataset[5] consists of 7974 high-resolution images in the train-val set with 26 labels at the L3 label hierarchy, taken from 182 different drive sequences. To create IDD-mini out of this dataset, we resize the images such that the largest dimension is downsampled to 720 while preserving the aspect ratio, and use the 16 labels from the L2 hierarchy of the original dataset. We subsample the number of images from the dataset by a factor of 4, uniformly across the drive sequences in such a way that the resultant split gives us the same proportion of labels as the full version of the dataset. The train set contains 1794 training images and 253 validation images.

IDD-lite The major aim of having IDD-lite, in addition to IDD-mini, is to enable very quick prototyping of semantic segmentation models which, we believe, is very essential for demonstration or teaching purposes in settings with resource limitations. Following a similar technique as explained above, we subsample the dataset by a factor of 10, which gives us 673 training and 110 test images. We rescale the largest dimension to 320 while preserving the aspect ratio of the image while using the L1 hierarchy with 7 coarse labels. This also reduces the required disk space to store the dataset from 18GB for IDD to <1.5GB for IDD-lite, which helps in optimizing the storage footprint.

While we provide training and validation splits along with the IDD-mini and IDD-lite datasets, we do not propose a separate test set different from the IDD test set which consist of 2029 images at the original resolution. We believe that this provides the models trained on different datasets with a common platform for bench marking. We hope that this will encourage the research community to come up with innovative architectures or algorithms for structure learning or

[5] https://idd.insaan.iiit.ac.in/

semi supervised learning to train models on such standard smaller scale labeled datasets, but still match the performance obtained by training on bigger datasets.

Label Statistics From Figure 2, it is shown that the mini and lite versions of IDD follow the same distribution as the original dataset, following the technique we used to subsample the dataset. The proportion of pixels corresponding to categories like *Road* and *Building* occupy a large fraction of the total annotated pixels, while there is also sufficient representation for smaller classes like *vehicle* and *traffic signs*. The total absolute number of annotated pixels (in log scale) is given in Figure 3, to show that the number of pixels in IDD-mini, IDD-lite are an order less than that of the original dataset.

Comparison with other Datasets Comparison to another large scale and widely used dataset, Citycapes [5], is also presented in Figure 2. Cityscapes consists of 2975 training images and 500 validation images at a uniform resolution of 1024×2048, with images taken from various cities and weather conditions. However, one major advantage that our datasets offer compared to cityscapes is that IDD-mini and IDD-lite contain scenes from more unstructured environments, with images captured from complex traffic and driving situations. Furthermore, the comparison from Figure 2 shows that on most categories, the smaller datasets match Cityscapes on the proportion of the pixels.

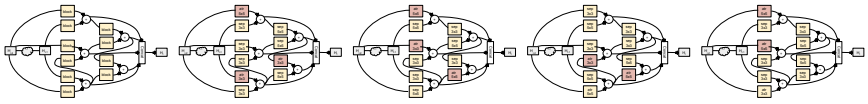

Fig. 4: To the left is Conv-module skeleton. From there to the right are the ERFNet modified models with Conv-module Cell structures named as A*, B*, C*, D* *(Best visualized when zoomed).*

3 Architecture Search

In this section, we demonstrate how IDD-lite and IDD-mini datasets can be useful for architecture search with limited resources. Neural Architecture Search [13] (NAS) on tasks like dense semantic segmentation need thousands of iterations GPU-days for convergence. Architecture search can be computationally very intensive as each evaluation typically requires training a neural network. Therefore, it is common to restrict the search space to reduce complexity and increase efficiency of architecture search. More recent papers on architecture search have shifted to searching the repeatable cell structure, while keeping the outer network level structure fixed by hand. This strategy needs less number of GPU hours and can conduct large experiments in constrained time. This would be a major advantage in resource constrained environments. Currently, this strategy is limited to tasks like image classification where small scale datasets are available but not

for semantic segmentation due to lack of such small scale datasets. We propose that IDD-mini, IDD-lite can be used to do architecture search in resource efficient way for semantic segmentation task.

We consider a scenario where we need to come up with best architecture with limited resource budget. We even explicitly consider a scenario where we can only afford fixed set of parameters for the architecture. Now, we need to find the best performing architecture among them. We conduct two experiments and show that architecture search results conducted on IDD-lite with custom ERFNet model actually translates to different domain like Cityscapes. Thus, we are also exploring generalizability of architectural search through such experiments.

Identifying Optimal Cell Structure Our aim is to find best architecture in custom designed architectural space using IDD-lite. Also, we show that the results correlate to Cityscapes dataset. We use ERFNet [14] outer network level structure as the basis for our model. Within this structure we replace non-bottleneck layer proposed in original ERFNet with custom structure with architecture skeleton(Conv-module) as shown in Figure 4.

Layer	Type
1	**Downsampler block**
2	**Downsampler block**
3-5	**3 x Conv-module**
5-7	**2 x Conv-module**
8	**Downsampler block**
9-16	**8 x Conv-module**(dilated)
17	**Deconvolution**(upsampling)
18-19	**2 x Conv-module**
20	**Deconvolution**(upsamling)
21-22	**2 x Conv-module**
23	**Deconvolution**(upsampling)

Layer	Type
1	**Downsampler block**
2	**Downsampler block**
3	**1 x Conv-module**
4	**1 x Conv-module**
5	**Downsampler block**
6	**Conv-module**(dilated 2)
7	**Deconvolution**(upsampling)
8	**1 x Conv-module**
9	**Deconvolution**(upsamling)
10	**1 x Conv-module**
11	**Deconvolution**(upsampling)

Table 2: Modified version(left) and Compressed version (right) of ERFNet architecture that is used to run experiments on Cityscapes dataset and IDD-lite respectively.

Each of the block in this Conv-module(skeleton) is filled from a set of 1 atrous 3×3 layer, 3 separable 5×5 layers, 2 atrous 5×5 layers, 4 seperable 3×3 layers, to ensure that we have same number of parameters overall. This forms the search space for architecture search. We use architectures given in Table 2 to conduct experiments on Cityscapes dataset and IDD-lite respectively.

4 Experiments and Results

Semantic Segmentation Performance Benchmarking In this section, we benchmark the results of the proposed datasets on two state of the art architectures used for semantic segmentation, DRNet [18] and ERFNet [14]. More details regarding these networks are present in [16], which we do not present here again in the interest of space. For DRNet, we use a ResNet-18 backbone(*drn-d-22*). We

Dataset	#L	Val. Res.	mIoU ERFNet [14]	mIoU DRN [18]
CS [14]	20	512×1024	71.50	68.00
IDD [16]	26	512×1024	55.40	52.24
IDD-mini	16	480×640	57.91	53.31
IDD-lite	7	128×256	66.14	55.03

(a)

Models	IoU (CS)	Params	IoU (IDD-lite)
ERFNet	70.45	2038448	53.975
D*	68.55	547120	52.01
DG2*	65.35	395568	50.71
DG4*	61.42	319792	48.88
DG8*	59.15	281904	46.40

(b)

Table 3: (a) Performance (in mIoU) of the proposed datasets on semantic segmentation architectures ERFNet and DRN-d-22. Note that *val. res.* corresponds to the validation resolution for each dataset, which is obtained by cropping and resizing the original images from Table 1, #L is the number of trainable classes in that dataset. (b) Depthwise Separable Convolution, Groups on ERFNet Architecture tested over IDD-lite dataset using Compressed ERFNet from Table 2.

Models	IoU(CS)	IoU(IDD-lite)
A*	64.54	58.15
B*	59.21	56.93
C*	55.96	55.46
D*	52.35	53.64

(a)

	IDD-lite	IDD-mini	IDD
ERFNet	1.12	5.99	10.60
DRNet-18	56.61	78.47	95.94

(b)

Table 4: (a) Custom Cell architecture on compressed ERFNet tested over IDD-lite dataset correlate with the same tests on Cityscapes Dataset with modified ERFNet from Table 2. (b) Inference time (in sec.) of different semantic segmentation models on various versions of IDD on Raspberry Pi 3B.

take mIoU (mean intersection over union) as the performance metric for all our experiments.

The models ERFNet and DRNet-18 were trained using the resolution depicted in Table 1 and validated using the resolution shown in Table 3(a). The models achieve an mIoU of 57.91% and 53.31% on IDD-mini using ERFNet and DRNet-18 respectively. Similarly, IDD-lite gives mIoU values of 66.14% on ERFNet and 55.03% on DRNet respectively.

From Figure 5, it is can be seen that IDD-lite dataset gives reasonably good mIoU results with just 15-20 minutes of training within 4GB GPU memory *(Best visualized when zoomed)*. We also note that while models trained on such datasets cannot directly be employed in state of the art semantic segmentation applications, they will nevertheless be very useful in for teaching or workshop purposes in cases with limited technical support and overall resource availability.

Results on architecture search Here, we present results on experiment to identify optimal cell structure and experimental correlation of IDD-lite and Cityscapes as mentioned in section 3.

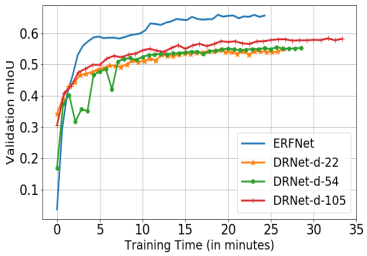

Fig. 5: Training time (x-axis) vs. Validation mIoU (y-axis) plot for IDD-lite. Note that with only 15 minutes of training on 1 GPU using only 4GB, the model obtains >50% mIoU

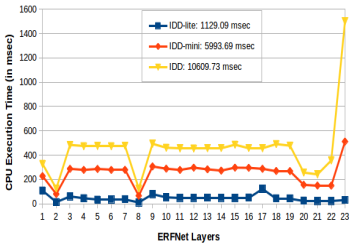

Fig. 6: Runtime statistics per layer for ERFNet model on all three datasets (IDD, IDD-mini and IDD-lite). Total run time for each dataset is mentioned in the legend *(Best viewed when zoomed)*.

Implementation details The best cell structure identified using architecture search gave 64.54% IoU (Model A) on Cityscapes dataset. We also take 3 models (B, C, D) from the pool of possible Conv-modules and verify if the performance on smaller dataset match with that on Cityscapes. The results are presented in Table 4(a). It is to be noted that though the architecture search was conducted on IDD-lite, we are able to get best performing architecture for a different domain like Cityscapes.

Correlation to Efficient Segmentation Models There have been lots of interest in efficient CNN module designs that have lower compute needs, while still achieving good prediction accuracies [7,19]. [15] reports results with architecture variations of ERFNet named as D*, D2*, D4*, D8* on Cityscapes. These correspond to the usage of depthwise separable convolutions instead of the bottleneck modules of ERFNet along with grouping parameter on the 1x1 convolution. We conduct the same experiments on IDD-lite dataset with compressed ERFNet architecture (Table 2) and show that our results correlate with [15]. The results are presented in Table 3(b).

Models for Raspberry Pi In order to make deep learning models scalable for real time application in resource constrained environments, factors such as real time performance, feasible cost of the hardware and low power consumption are essential. Hence, we provide the benchmarking values of segmentation on Raspberry Pi. This device is widely available as a single board compute platform which comes at an affordable cost, apart from being customizable and energy efficient.

More specifically, we chose Raspberry Pi 3B as the deployment hardware device for our semantic segmentation models. The device contains 1 GB RAM, and has a 1.2GHz Quad-Core 4XARM Cortex-A53 CPU. We tested ERFNet and DRNet-18 networks on Raspberry Pi to calculate the inference time on the validation datasets at various resolutions. Table 4(b) shows the inference time of

different semantic segmentation models at various resolutions on our validation datasets.

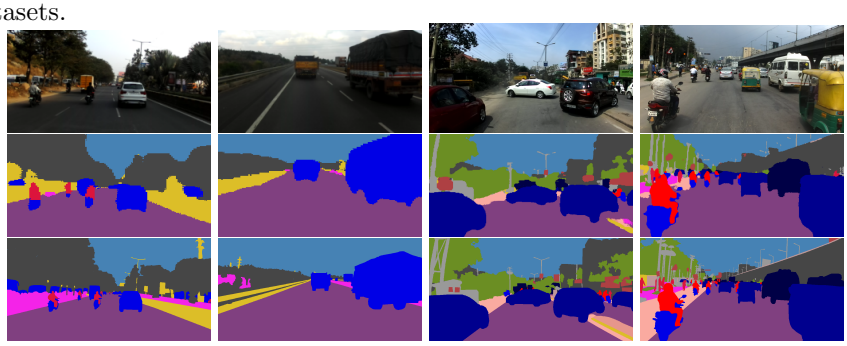

Fig. 7: Qualitative examples of ERFNet model run on IDD-mini, IDD-lite dataset. From top to bottom - image, prediction and ground truth. First two columns correspond to results from IDD-mini and last two columns are for IDD-lite *(Best viewed when zoomed)*.

Figure 6 shows the run time information of each layer defined in the ERFNet architecture. Although the IDD and the IDD-lite datasets have equal trends, IDD-lite time is significantly lesser, in addition to being uniformly consistent across layers. This further reinforces our proposition that such a dataset can add more value to quick prototyping and help move towards real time deployment of segmentation models.

5 Conclusion

We propose two small scale datasets, IDD-mini and IDD-lite, to address some of the relevant issues in training semantic segmentation models on resource constrained environments. We show that these carefully designed datasets give decent qualitative and quantitative results enabling fast prototyping on low resource hardware and hugely reducing the training and deployment costs. We also demonstrate the usefulness of such small scale datasets in performing architecture search by showing that the parameters obtained using smaller network on these datasets actually translate to larger network with high resolution images.

References

1. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters **30**(2), 88–97 (2009)
2. Cai, H., Chen, T., Zhang, W., Yu, Y., Wang, J.: Efficient architecture search by network transformation. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
3. Chen, L.C., Collins, M., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J.: Searching for efficient multi-scale architectures for dense image prediction. In: Advances in Neural Information Processing Systems. pp. 8713–8724 (2018)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: 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 (2018)

5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
6. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361. IEEE (2012)
7. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
9. LeCun, Y.: The mnist database of handwritten digits. http://yann. lecun. com/exdb/mnist/ (1998)
10. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A., Fei-Fei, L.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. arXiv preprint arXiv:1901.02985 (2019)
11. Liu, H., Simonyan, K., Vinyals, O., Fernando, C., Kavukcuoglu, K.: Hierarchical representations for efficient architecture search. arXiv preprint arXiv:1711.00436 (2017)
12. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
13. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. arXiv preprint arXiv:1802.03268 (2018)
14. Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. IEEE Transactions on Intelligent Transportation Systems **19**(1), 263–272 (2018)
15. Vallurupalli, N., Annamaneni, S., Varma, G., Jawahar, C., Mathew, M., Nagori, S.: Efficient semantic segmentation using gradual grouping. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 598–606 (2018)
16. Varma, G., Subramanian, A., Namboodiri, A., Chandraker, M., Jawahar, C.: IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE (2019)
17. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
18. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 472–480 (2017)
19. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6848–6856 (2018)
20. Zhong, Z., Yan, J., Wu, W., Shao, J., Liu, C.L.: Practical block-wise neural network architecture generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)