Training Efficient Semantic Segmentation CNNs on Multiple Datasets

Marco Leonardi, Davide Mazzini(✉), and Raimondo Schettini

University of Milano - Bicocca, Milano, Italy
m.leonardi6@campus.unimib.it, {davide.mazzini,schettini}@unimib.it

Abstract. In the past few years, various datasets for semantic segmentation have been presented. However, dense per-pixel groundtruths are difficult and expensive to obtain, therefore every single dataset contains only a subset of the semantic classes required to fully understand outdoor environments for real-world applications, e.g. autonomous or assisted driving. In this work, we investigate a simple approach to modify semantic segmentation CNNs in order to train them on multiple datasets with heterogeneous groundtruths. We trained and tested six efficient Deep CNN models on three datasets with different types of annotations such as generic objects, traffic signs and lane markings. Experiments show that the networks are trainable with the implemented method even though it highlights the limit of current efficient architectures when dealing with heterogeneous and large datasets.

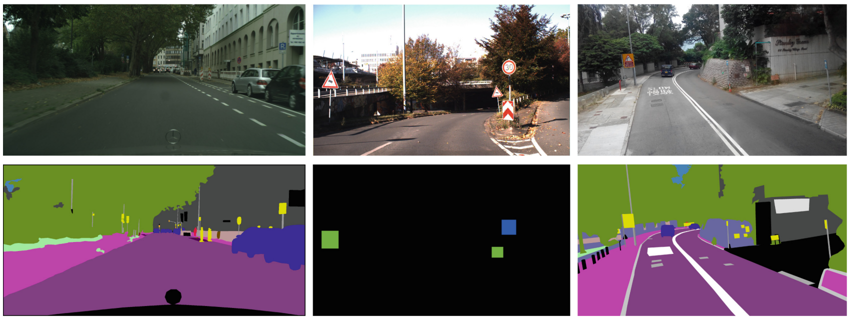
Keywords: Deep Convolutional Neural Networks · Semantic segmentation · Scene understanding

1 Introduction

CNN architectures for semantic segmentation can be classified in two categories: accuracy-oriented and efficiency-oriented methods. Former methods achieve high accuracy with very high computational costs whereas latter methods attain very high inference speed and low memory footprint with evident loss in accuracy. Most efficiency-oriented methods are fast enough to run in real-time on recent embedded devices [7,9–11]. This enables novel applications in fields like autonomous or assisted driving where a complete scene understanding is of paramount importance. In such scenarios the perception model should be able to deal with a high number of heterogeneous classes in order to approach the real-world complexity. However, efficiency-oriented CNNs in literature are usually benchmarked on small datasets with a limited set of classes. In this work we modify six efficient Deep CNN models in order to enable them to deal with multiple datasets during training. We tested the networks on three datasets: Cityscapes, GTSDB and Vistas. GTSDB in particular has only bounding-box annotations, thus we investigate an approach, inspired by [12], to train semantic segmentation networks on dense per-pixel groundtruth along with bounding boxes. We assess the performance of efficient CNN models focusing on three aspects: tracking model behavior when it deals with an increasing number of datasets; assessing the impact of decoder structure; and quantify the influence of pretraining.

2 Datasets

In the following sections we describe the datasets and metrics adopted to carry out our experiments (Fig. 1).



a) Cityscapes b) GTSDB c) Mapillary Vistas

Fig. 1. Sample images from *Cityscapes*, *GTSDB* and *Mapillary Vistas* together with their associated groundtruth maps. GTSDB annotations are very sparse and include traffic signs only. Mapillary Vistas exhibit a richer classes set compared to Cityscapes, e.g. including lane markings.

Cityscapes [2] is a dataset of urban scenes images with semantic pixelwise annotations. It consists of 5000 finely annotated high-resolution images (2048×1024) of which 2975, 500, and 1525 belong to train, validation and test sets respectively. Annotations include 30 different object classes but only 19 are used to train and evaluate models. With its basic classes hierarchy, Cityscapes it is not fully comprehensive of important annotations like lane markings or the different traffic signs. Such categories are instead annotated on other datasets presented in this Section.

German Traffic Sign Recognition Benchmark [5] (GTSDB) is a dataset for traffic sign localization and classification. It contains 900 images of street scenes acquired with different illumination and weather conditions. 43 classes of pole traffic signs are annotated with bounding-boxes. To our knowledge there is no publicly available dataset with per-pixel dense semantic segmentation annotations of traffic signs. GTSDB contains bounding-box annotations for traffic signs and no other annotations for different objects in the scene. In the next Section we introduce a technique, inspired by [12], to modify all the architectures in order to deal with bounding-box annotations.

Mapillary Vistas [13]**.** It is one of the biggest datasets for urban scene understanding publicly available to date. It is composed of 25000 images from different locations all over the world. It contains 18000, 2000 and 5000 images for training, validation and test respectively. Images have been acquired with a high variability of light and weather conditions. Annotations consist of 66 finely-annotated semantic classes. Since this dataset contains a large number of object categories, we employ it in two different settings: In the first setting we utilize only four classes relatives to lane markings (road, lane markings, crosswalk plain and crosswalk). In the following Sections, we will call it *Lane Markings* dataset, even though it is actually a subset of the Vistas dataset. In the second setting, we will make use of all the classes included in the dataset naming it *Vistas* in the next Sections.

2.1 Evaluation Metrics

Semantic segmentation quality has been evaluated by means of the *mean of class-wise Intersection over Union (mIoU)* which is computed, following [4], as the class-wise mean of the intersection over union measure. Models' speed is computed in *Frame Per Second (FPS)*, defined as the inverse of time needed for our network to perform a single forward pass on a single NVidia Titan Xp GPU.

3 Efficient Semantic Segmentation Networks

In this Section, we introduce and describe the efficient models for semantic segmentation considered in this work. We outline a brief overview of the peculiar characteristics of each architecture:

ENet [14] adopts an encoder-decoder design with 28 stacked bottleneck layers. Its efficiency is due to different factors, mainly the use of asymmetric convolutions. Different works investigated the use of asymmetric convolutions both from a theoretical point-of-view [1,17] and in a practical setting [18,19]. The main idea is to spatially decompose 3×3 convolutional filters into 3×1 followed by 1×3 filters. ENet introduced other design patterns that have become common in other efficient architectures: e.g. early downsampling and dilated convolutions [20].
GUNet [8] makes use of a multi-resolution encoder to exploit at the same time low-resolution structures and high-resolution details. The two resolutions are processed by different branches of the network which shares the same weights. Multi-resolution features are fused by means of a Fusion Module. The decoder part consists of a Guided Upsampling Module to efficiently output the prediction map improving the predictions near objects' boundaries.

ERFNet [15] employs an encoder-decoder structure inspired by ENet where the main building block is modified to increase model accuracy. It introduces a new block called Non-Bottleneck-1D and a fast downsampling strategy where the inner activations are spatially downsampled in the earlier part of the network to keep the majority of the computational burden for the deepest stages.

iGUM-Net [10] is an encoder-only architecture. In [10] is introduced an improved version of the GUM module, first presented in [8], that allows removing the decoder part of the network without any loss in segmentation quality. The encoder is inspired by ERFNet [15] with early downsampling and a modified Non-Bottleneck-1D Module.

Edanet [7] is heavily based on ERFNet. It makes use of asymmetric convolutions and dilated residual blocks. Subsampling is performed with ENet [14] downsampling blocks. The main difference with other architectures is the adoption of dense connectivity patterns between residual blocks. Like [10], Edanet is an encoder-only architecture but without any refinement module to improve high-resolution boundaries.

ESPNet [11] is an encoder-decoder architecture with concatenated skip connections similar to U-Net [16]. The peculiar trait of ESPNet is the introduction of the Efficient Spatial Pyramid (ESP) module. It consists in a *reduce-split-transform-merge* pattern. It is composed of point-wise convolutions to project features to a low-dimensional space, a pyramid of dilated convolutions to perform large receptive-field operations and point-wise convolutions to merge the computational paths (Fig. 2).

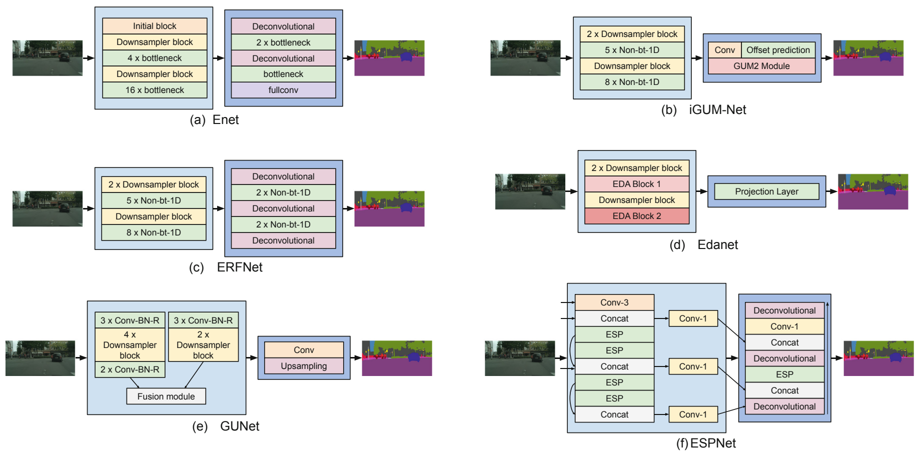


Fig. 2. Schematic view of the considered efficient CNN architectures. Light blue blocks represent encoders whereas darker blocks represents models' decoders. (Color figure online)

4 A Simple Method to Train on Heterogeneous Datasets

We modified each network in order to handle multiple heterogeneous datasets simultaneously during training. Our setup is inspired by [12] but presents some differences. We address the problem in a straightforward way, adopting one encoder and multiple decoders: one for each dataset. This is referred as *flat classification* in [12]. The main advantage of adopting such architectural choice is that we do not have to take care of the relations between the different dataset hierarchies. Each decoder is trained with an independent softmax on that specific dataset's classes, whereas the encoder is trained jointly. We made only one experiment towards modifying this general structure. We tested two different configurations modifying the network layers trained on the specific dataset: the whole decoder vs only a single final convolution. All the details of these experiments are described in Sect. 5.2.

Dealing with Bounding Boxes Annotations. All models considered in our evaluation have been conceived to produce a dense per-pixel prediction map of scene semantic segmentation. They are designed to be trained and tested with dense per-pixel annotations. However, the GTSDB dataset has only bounding boxes annotations. To overcome this issue we implemented a training setup, inspired by [12], that allows training semantic segmentation models with GTSDB annotations. A schematic view of the whole pipeline is depicted in Fig. 3. A per-pixel groundtruth is shaped in two steps: first, the GTSDB bounding-boxes are converted into pseudo per-pixel groundtruths, i.e. squared areas corresponding to the bounding box size and location. As a second step, they are merged with the prediction from the Cityscapes branch corresponding to the *generic* traffic sign class. The obtained groundtruth is then employed in the usual way to train the network branch with a per-pixel cross entropy loss.

Dealing with Heterogeneous Cardinalities. Each dataset adopted in our experiments has very different cardinality from each other. In order to balance the impact of each dataset when training CNNs, we replicate the datasets with lower dimension to the cardinality of the dataset with the maximum number of images, see Sect. 2 for details. We set the number of training iterations to a fixed number, corresponding to 150 Cityscapes epochs, to make the network see the same number of images independently of the dataset choice. Furthermore, to balance the information gain from each dataset, we weight the loss generated by each model branch with a learned variable parameter.

Architecture Alternatives. Many networks considered in this work have a complex decoder composed of multiple Conv-Batchnorm-Relu blocks followed by upsampling or transposed convolution operators. Duplicating the decoder part leads in computationally heavier models. For this reason, we also evaluate an alternative architectural choice in which only the last convolutional layer is repeated for each dataset instead of the whole decoder.

Training Recipe. We trained all models minimizing the cross entropy loss between the ground-truth and the predicted classes for each pixel. We employed

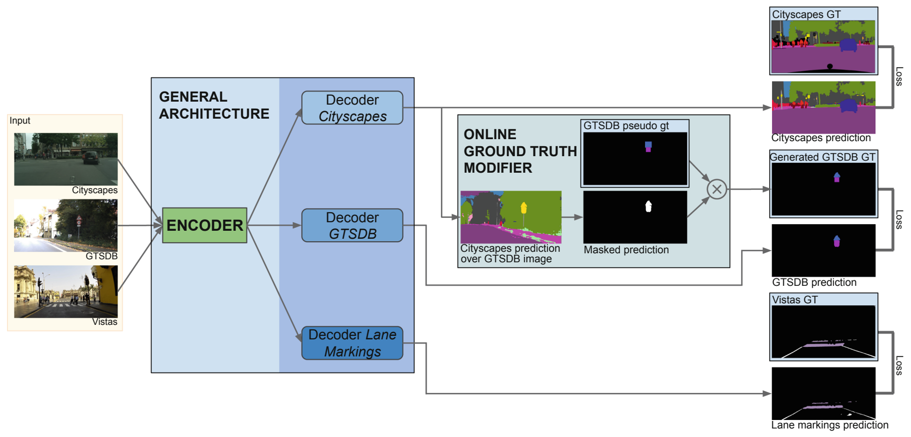


Fig. 3. Overview of the training setup to train models on multiple datasets simultaneously. Every model has been modified to follow this general architecture. The online ground truth modifier allows to train the GTSDB decoder with bounding box annotations.

ADAM as stochastic optimizer [6] with a base learning rate of 5×10^{-4}. We adopted a polynomial decay learning rate policy defined as follows: $LR(epoch) = \left(1 - (epoch/total\ epochs)^{0.9}\right) * LR_0$ where *epoch* is the 0-based index of the actual epoch, LR_0 is the initial learning rate, and *total epochs* is the total number of epochs for the training process. The batch size is set to six for all the experiments. Furthermore, we employed early stopping technique against overfitting, analyzing the mean intersection over union on the validation set for every epoch. The only preprocessing step consists in normalizing the input image by subtracting the mean and dividing by the standard deviation computed on Imagenet [3].

Data Augmentation. We adopted classical augmentation techniques during training. In particular, since images from different datasets have different sizes, we scale input images such as the longest axis is 1024 for the width or 512 for the height. We apply random scale augmentation sampling the scaling factor from a uniform distribution with interval $[0.5, 2]$.

5 Experiments

In order to assess the behaviour of the considered models on multiple heterogeneous datasets, we investigated three research directions that led to three related sets of experiments. In the first set, we benchmark CNN models and track their behaviour when dealing with an increasing number of datasets. In the second and third set of experiments, we evaluate respectively the impact of the decoder structure and the influence of pretraining.

5.1 Scaling up with Dataset Complexity

We evaluate each model in four experimental settings. In each set a dataset is added to the training set with new images and the set of object categories is extended:

1. **Baseline.** The baseline experiment consists of training on Cityscapes dataset: it comprises 19 semantic classes. CNN architectures have been evaluated on this dataset with no modification to the original structure. Unique images: 2975. We adopt this dataset as our baseline for two main reasons: first, Cityscapes is a well-known benchmark dataset for semantic segmentation methods and it is widely adopted in literature. The performance of all methods considered in this work has been evaluated on Cityscapes in the original papers. Second, it includes only the basic classes to perform a complete scene understanding on urban environments whereas the other datasets consist of a superset of Cityscapes i.e. Vistas, or they include only a specific subset of classes with finer annotations i.e. traffic signs or lane markings.
2. **Cityscapes + GTSDB.** In this experiment, models have to predict simultaneously Cityscapes and GTSDB classes for a total of 62 classes (i.e. 19 from Cityscapes + 43 from GTSDB). In this case, the architectures have been modified to include two decoders. Unique images: 3490. We included this dataset before Lane Markings or Vistas for reasons bound to the specific application. Traffic signs are a fundamental element when dealing with scene perception for autonomous driving.
3. **Cityscapes + GTSDB + Lane Markings.** In this setup CNN models predict 62 classes plus 4 classes form Lane Markings dataset. The number of decoders for each architecture in this setting is three. Unique images: 21490.
4. **Cityscapes + GTSDB + Vistas.** In this experiment, the models are required to predict all the 104 classes from all datasets. The number of decoders is three (since Vistas includes Lane Markings). Unique images: 21490.

In Fig. 4 is reported a plot of the mean Intersection over Union versus the number of datasets. With this visualization, we want to track the mean performance of each model when dealing with increasing images and/or classes cardinality. The general tendency is a degradation of the overall performance as the problem complexity increases. The considered models are not designed for highly complex problems, and since their limited number of parameters, their capacity is presumably close to saturation. Models behaviour with respect to the Lane Markings dataset seems a counter example to the general trend. This behaviour is plausible since the visual appearance of lane markings is very distinguishable from other semantic classes.

The same set of experiments can be visualized in Fig. 5 from a different point of view. It represents the mIoU computed on the Cityscapes validation set with Cityscapes classes only. By looking from this perspective we want to assess models behaviour on the original dataset when trained with additional images and extra classes. Interestingly the considered models exhibit better performance on

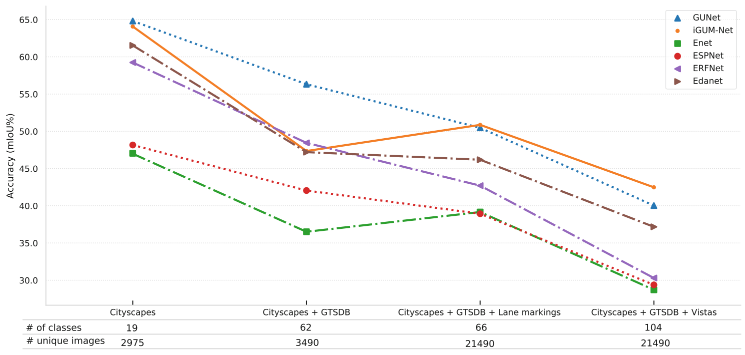


Fig. 4. Average mIoU measured on the joint datasets. On the x axis the datasets considered to train and test the models along with the resulting number of classes and unique images.

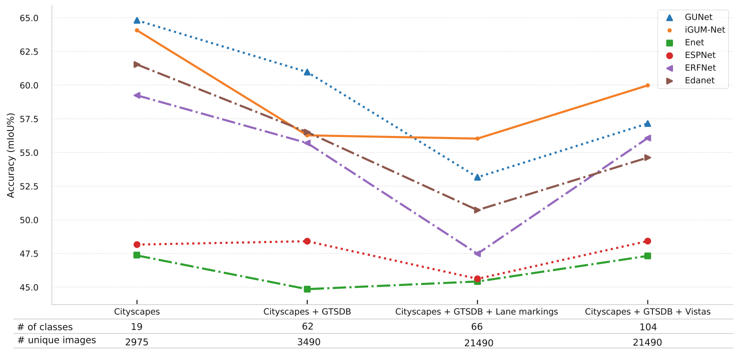


Fig. 5. mIoU on the Cityscape validation dataset versus the datasets employed in training along with the resulting number of classes and unique images.

Cityscapes + GTSDB + Vistas experiment than in previous experiments. Even if it is more complex with higher number of classes, this behaviour is plausible since the Vistas dataset includes a large superset of the Cityscapes class set.

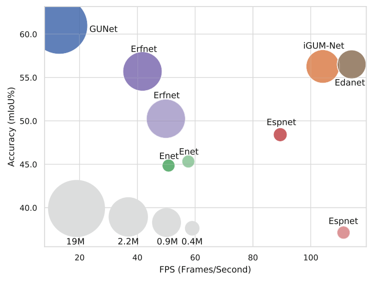
In Table 1 we report the comprehensive set of results of the experiments introduced in this Section and shown in Figs. 4 and 5. We also included results from [12] which is the only work in the state of the art that experimented with a similar setup, main differences are: their network structure is specifically tailored to handle the classes hierarchy of the joint datasets; there is a difference in the cardinality of Cityscapes and Vistas classes. For Cityscapes they use more than 19 classes. For Vistas they report performance over all classes while our performance regards only the additional classes w.r.t Cityscapes.

Table 1. Benchmarks of different models trained and tested on different datasets (* number of classes is higher)

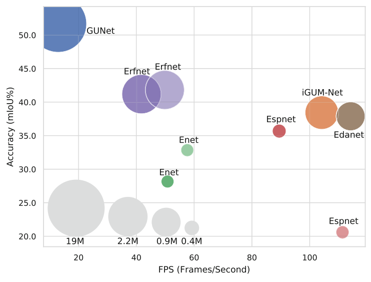
	mIoU								
Tested on	Cityscapes	Cityscapes	GTSDB	Cityscapes	GTSDB	Lane markings	Cityscapes	GTSDB	Vistas
# Classes	19	19	43	19	43	4	19	43	48
Meletis et al.	N/A	N/A	N/A	N/A	N/A	N/A	57.3*	41.5	**31.9***
GUNet	**64.8**	**61**	**51.7**	53.2	43.6	**54.7**	57.2	48.5	14.4
iGUM-Net	64.1	56.3	38.4	**56**	**49**	47.6	**60**	**50.6**	16.8
Enet	47.3	44.9	28.2	45.4	30.2	41.9	47.3	27.4	11.4
ESPNet	48.2	48.4	35.7	45.6	31.7	39.5	48.4	27.3	12.4
ERFNet	59.2	55.7	41.2	47.5	34.5	46.1	56.1	24.3	10.6
Edanet	61.5	56.5	37.9	50.7	43	44.9	54.6	41.7	15.2
Trained on	Cityscapes	Cityscapes + GTSDB		Cityscapes + GTSDB + Vistas			Cityscapes + GTSDB + Vistas		

5.2 Multiple vs Single Decoder

Figure 6 shows a comparison of mIoU versus model complexity, expressed in Parameters and FPS, tested on Cityscapes and GTSDB datasets. All models benefit in term of speed from the adoption of a single decoder whereas the number of parameters does not visibly decrease. Enet is the only model that take advantage even in terms of accuracy from this architectural design. Some models are represented by a single point in Fig. 6 i.e. GUNet, iGUM-Net and Edanet. They have been conceived, in their original structure, with a lightweight decoder design composed by only a convolutional layer.



(a) Tested on Cityscapes



(b) Tested on GTSDB

Fig. 6. Multiple decoders (solid colors) vs single decoder (transparent colors): comparison between accuracy in terms of mean intersection over union and frame per second. Models have been trained jointly on Cityscapes and GTSDB and separately tested. Circle dimension reflects the number of model parameter.

5.3 Impact of Pretraining

It is well known from semantic segmentation literature that the overall model performance improves by adopting a pretrained encoder on Imagenet [8,10,15].

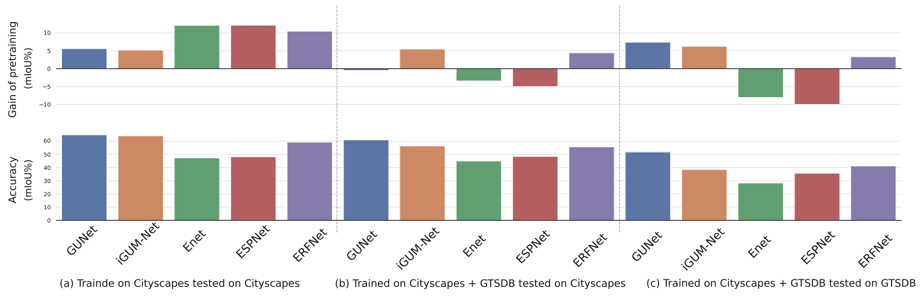


Fig. 7. mIoU on different combinations train and test datasets. Lower bars represent the non-pretrained baseline. Higher bars express the absolute gain when pretraining.

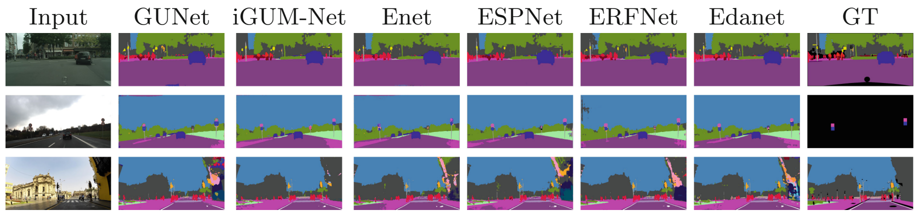


Fig. 8. Sample images from Cityscapes (first row), GTSDB (second row) and Mapillary Vistas (third row) datasets, together with their groundtruths.

For this reason, we want to investigate the impact of pretrained encoders on our experimental setup. In Fig. 7 we show a comparison on the use of a pre-trained encoder. As delineated in Fig. 7, the benefit obtained using the pre-trained encoder is effective only in the experiments trained solely on Cityscapes with an average increment in accuracy of 7%. On the contrary, models trained on both the Cityscapes and GTDSB improve on average by 0.2 on Cityscapes but decrease by −0.2 on GTSDB (Fig. 8).

6 Conclusions

We investigated a simple approach to train semantic segmentation CNNs on multiple datasets with heterogeneous groundtruths. We assessed the performance of six efficient semantic segmentation networks trained with such approach. We tracked the behavior of all models across different experiments with increased training datasets. Experimental results shows that, by training on increasingly large and diverse datasets, performance of all models decreases as a general trend. These results show that, actual efficient models, are probably not able to deal with more complex datasets and classes hierarchies. We also tested two different decoder variants to deal with multiple datasets and evaluated the impact of pretraining on models performance. Results of these experiments suggest that their influence on the overall pipeline is beneficial in terms of speed but not effective in terms of accuracy gain. As possible future work, it would be interesting to study the benefits and the drawbacks on the use of more complex architectures. Future experiments will also investigate the different elements involved in employing multiple heterogeneous datasets (e.g. diverse groundtruths, classes and images distributions) in order to disentangle the contribution of each factor.

Acknowledgments. This work was supported by TEINVEIN, CUP: E96D1700011 0009 - Call "Accordi per la Ricerca e l'Innovazione", cofunded by POR FESR 2014–2020. We gratefully acknowledge the support of NVIDIA Corporation with the donation of a Titan Xp GPU used for this research.

References

1. Alvarez, J., Petersson, L.: Decomposeme: simplifying convnets for end-to-end learning. arXiv:1606.05426 (2016)
2. Cordts, M., et al.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on CVPR (2016)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
4. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. IJCV **88**(2), 303–338 (2010)
5. Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: the German traffic sign detection benchmark. In: International Joint Conference on Neural Networks, No. 1288 (2013)
6. Kinga, D., Adam, J.B.: A method for stochastic optimization. In: ICLR (2015)
7. Lo, S.Y., Hang, H.M., Chan, S.W., Lin, J.J.: Efficient dense modules of asymmetric convolution for real-time semantic segmentation. arXiv:1809.06323 (2018)
8. Mazzini, D.: Guided upsampling network for real-time semantic segmentation. In: British Machine Vision Conference (BMVC) (2018)
9. Mazzini, D., Buzzelli, M., Pau, D.P., Schettini, R.: A CNN architecture for efficient semantic segmentation of street scenes. In: 8th ICCE-Berlin, pp. 1–6. IEEE (2018)
10. Mazzini, D., Raimondo, S.: Spatial sampling network for fast scene understanding. In: Proceedings of the IEEE Conference on CVPR Workshops (2019)
11. Mehta, S., Rastegari, M., Caspi, A., Shapiro, L., Hajishirzi, H.: ESPNet: efficient spatial pyramid of dilated convolutions for semantic segmentation. arXiv:1803.06815 (2018)
12. Meletis, P., Dubbelman, G.: Training of convolutional networks on multiple heterogeneous datasets for street scene semantic segmentation. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1045–1050. IEEE (2018)
13. Neuhold, G., Ollmann, T., Rota Bulo, S., Kontschieder, P.: The mapillary vistas dataset for semantic understanding of street scenes. In: Proceedings of the IEEE ICCV, pp. 4990–4999 (2017)
14. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: ENet: a deep neural network architecture for real-time semantic segmentation. arXiv:1606.02147 (2016)
15. Romera, E., Alvarez, J.M., Bergasa, L.M., Arroyo, R.: ERFNet: efficient residual factorized convnet for real-time semantic segmentation. IEEE Trans. Intell. Transp. Syst. **19**(1), 263–272 (2018)

16. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
17. Sironi, A., Tekin, B., Rigamonti, R., Lepetit, V., Fua, P.: Learning separable filters. IEEE Trans. PAMI **37**(1), 94–106 (2015)
18. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-ResNet and the impact of residual connections on learning. In: AAAI, vol. 4, p. 12 (2017)
19. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE CVPR, pp. 2818–2826 (2016)
20. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)