2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Training of Convolutional Networks on Multiple Heterogeneous Datasets for Street Scene Semantic Segmentation

Panagiotis Meletis and Gijs Dubbelman

***Abstract*— We propose a convolutional network with hierarchical classifiers for per-pixel semantic segmentation, which is able to be trained on multiple, heterogeneous datasets and exploit their semantic hierarchy. Our network is the first to be simultaneously trained on three different datasets from the intelligent vehicles domain, *i.e.* Cityscapes, GTSDB and Mapillary Vistas, and is able to handle different semantic level-of-detail, class imbalances, and different annotation types, *i.e.* dense per-pixel and sparse bounding-box labels. We assess our hierarchical approach, by comparing against flat, non-hierarchical classifiers and we show improvements in mean pixel accuracy of 13.0% for Cityscapes classes and 2.4% for Vistas classes and 32.3% for GTSDB classes. Our implementation achieves inference rates of 17 fps at a resolution of 520 x 706 for 108 classes running on a GPU.**

I. INTRODUCTION

Semantic classification is a key task in the perception sub-system of an autonomously driving vehicle [1]. The segmentation task, posed as per-pixel classification, has seen great progress in the past years [2] due to deep learning techniques. However, two critical challenges that still need to be addressed are: 1) to utilize as much and diverse training data as possible, and 2) to increase the number of recognizable classes from a few dozens to virtually anything that a scene can contain.

In this work, we take steps towards solving both challenges and we present a method that leverages multiple *heterogeneous* datasets, *i.e.* datasets with different classes and annotation types, to train a fully convolutional network for per-pixel semantic segmentation. This approach facilitates better use of available datasets, thereby reducing annotation effort, and increases the number of classes that can be recognized. The datasets that we are using in the context of Highly Automated Driving (HAD) are Cityscapes [3], Mapillary Vistas [4], and GTSDB [5].

The first challenge, *i.e.* training for semantic segmentation with diverse annotations, is tackled in previous works [6], [7] by external components to the network, in order to generate pseudo per-pixel ground truth. Our method, in contrast, is self-inclusive and uses the networks' own outputs to refine non-compatible, diverse annotations for supervision.

The second challenge, *i.e.* to increase the number of recognizable classes, can be accomplished in two ways: 1) continue per-pixel annotating an existing dataset with the extra (sub)classes, e.g. [8], or 2) use existing auxiliary datasets only for the new (sub)classes. The first approach can be very costly for big datasets and mainly unnecessary, as a plethora of datasets with fine-grained (sub)classes exist (*e.g.* traffic sign types, car models, pedestrians). In our work, we research the second approach. For this, the heterogeneity, *i.e.* different label spaces and annotations types, of datasets poses challenges for combining them with traditional "flat", *i.e.* non-hierarchical, classifiers. Therefore, we propose the use of hierarchical classifiers, which explicitly take advantage of the semantic relationships between the datasets, and we compare against flat classifiers. Our hierarchy is comparable to [9], [10], but differs in the scalability it offers.

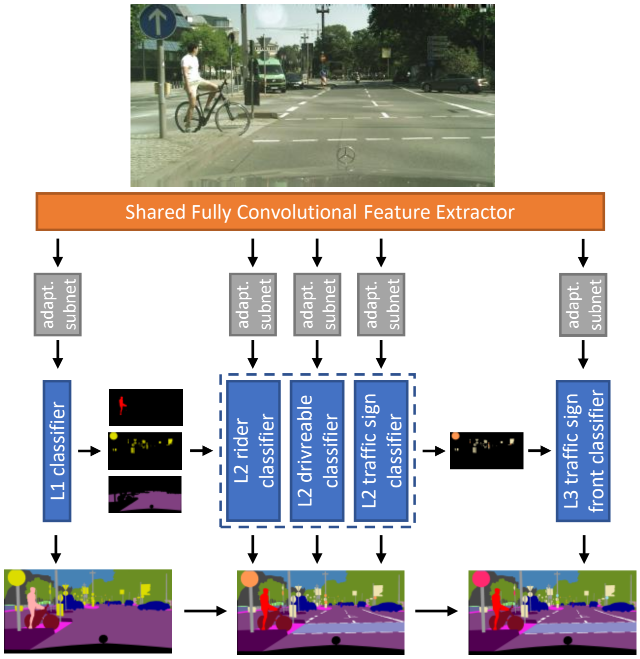

Fig. 1. Our hierarchical classification convolutional network during inference. The input image is transformed to a shared feature representation, which is connected to a hierarchy of classifiers though adaptation subnetworks. The Level-1 classifier outputs predictions for every pixel of the image, while each subsequent classifier infers only about its own set of classes. The output of all levels is combined to form the final fine-grained per-pixel segmentation.

In Sec. II, we describe the exact challenges that are addressed by our hierarchical approach. An example is combined training on Cityscapes and GTSDB. In that case, all classes of GTSDB are subclasses of the traffic sign class in Cityscapes. The straightforward approach of combining classes from both datasets in a conventional flat classifier is infeasible, since a traffic sign pixel cannot have different

Panagiotis Meletis (`p.c.meletis@tue.nl`) and Gijs Dubbelman (`g.dubbelman@tue.nl`) are with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 688099.

978-1-5386-4451-5/18/$31.00 ©2018 IEEE

labels depending on the dataset it comes from. This poses challenges for end-to-end training and inference for flat classifiers, which are addressed by our hierarchical classification scheme.

The fundamentals of our general hierarchical approach are provided in Sec. III and in Sec. IV we provide the specifics of our implementation. In Sec. V, we demonstrate the performance gain of hierarchical classifiers with the three *heterogeneous* datasets, over flat, non-hierarchical classifiers. Furthermore, we show that multi-dataset training of a common feature representation, using our proposed method, can improve performance across all datasets regardless of their structural differences.

To summarize, the contributions of this work to per-pixel semantic segmentation are:

- A methodology for combined training on datasets with disjoint, but semantically connected, label spaces.
- A modular architecture of hierarchical classifiers that can replace the classification stage in modern convolutional networks.

Our system implementation is made available to the research community [11]. Additionally, we provide our per-pixel annotations of the Cityscapes dataset for the GTSDB traffic sign subclasses, which we use for validation purposes, but is not required for training. We refer to this dataset as Cityscapes Extended throughout this paper.

II. Challenges from multiple dataset training

End-to-end supervised training on multiple datasets can face many challenges due to the structural differences of the datasets. The most important challenges can be categorized in the following groups:

Semantic level-of-detail: Each dataset is labeled with a set of semantic classes. In the naive flat classification approach, the output of the classifier will be the union of classes from all datasets. It is highly probable that the semantics of a class in one dataset contain, or are contained in, the semantics of a class of another dataset. If those classes are placed in the same level, as in the case of a flat classifier, a conflict of supervision happens, since some pixels belonging to the same semantic class will be labeled with different classes.

For our three datasets, this challenge appears in three cases: 1) Cityscapes defines its road class as: "part of ground on which cars usually drive", which includes lane markings, bicycle lanes, potholes, etc. In Vistas, these fine-grained subclasses are separately labeled, in addition to the a road class, leading to conflicts in the semantic level-of-detail of labels, 2) Cityscapes and Vistas contain one high-level traffic sign class, while GTSDB has 43 traffic sign subclasses, and 3) Cityscapes has only one rider class, while Vistas differentiates between three different rider subclasses. Introducing a hierarchy of labels, as shown in Fig. 2, effectively solves this challenge, and is discussed in more detail in Sec. III-A.

Annotation types: Semantic segmentation is by definition a per-pixel problem, thus supervision must be provided at the pixel level. Unfortunately, many existing datasets have bounding-box or per-image annotations, which are incompatible for per-pixel training. The direct approach to make them compatible would be to convert those annotations to masks. However, these masks will include pixels that don't belong to the object of interest, for example bounding boxes might include many non-relevant background pixels that will be assigned to the foreground class. Eventually, during training, supervision to the network will flow from incorrectly labeled pixels, leading to weights confusion.

In our case, Cityscapes and Vistas have per-pixel annotations, but GTSDB has only bounding box annotations. In order to include GTSDB during training, we propose a novel hierarchical loss, provided in III-D, which uniformly handles supervision from different annotation types.

Training sample imbalances: Batch-wise training suffers from class imbalances, especially when there are limited examples per batch. In our case, we face strong intra-dataset and inter-dataset imbalances. Imbalances between datasets happen due to the big different of annotated pixels, *i.e.* in the order of 10^3 (see Table I). Imbalances between the classes of the same dataset are common for street scene datasets, since most of the pixels belong to classes of big surfaces, like road and buildings. Our method deals with imbalances by placing classes with the similar order of examples in the same classifier and thus all classes have bigger probability to be represented in the same batch. This strategy is highly beneficial, as shown in Sec. V-E.

III. Training and inference on heterogeneous datasets using semantic hierarchies

In this Section, we describe the components of our general hierarchical classification methodology for an arbitrary number of *heterogeneous* datasets. These components provide solutions to the challenges of Sec. II and for each one we elaborate on the specific cases for our selected datasets. Our current experiments, detailed in Sec. V, are based on an implementation with a 3-level hierarchy using 3 datasets. The specifics of this implementation are provided in Sec. IV.

A. Semantic hierarchy of label spaces

Multiple dataset training requires a common label space for all selected datasets. We propose to combine individual label spaces into the common space, containing labels from all datasets, by a hierarchical manner into a semantic tree of labels. This approach solves any conflict in the semantic definition of labels, by introducing the necessary parent or intermediate nodes and/or grouping of existing labels.

Figure 2, depicts the 3-level hierarchy using all labels for the three selected datasets of this paper. The challenges that arise from combining these three label spaces were described in Sec. II and are solved as follows: 1) A new high-level driveable class is introduced to solve Cityscapes and Vistas road class semantic conflict, 2) A superclass of traffic signs and an intermediate node for differentiating Vistas *back* from *front* traffic signs are added, and 3) A rider superclass is introduced to include the Cityscapes rider class and the 3 Vistas rider subclasses.

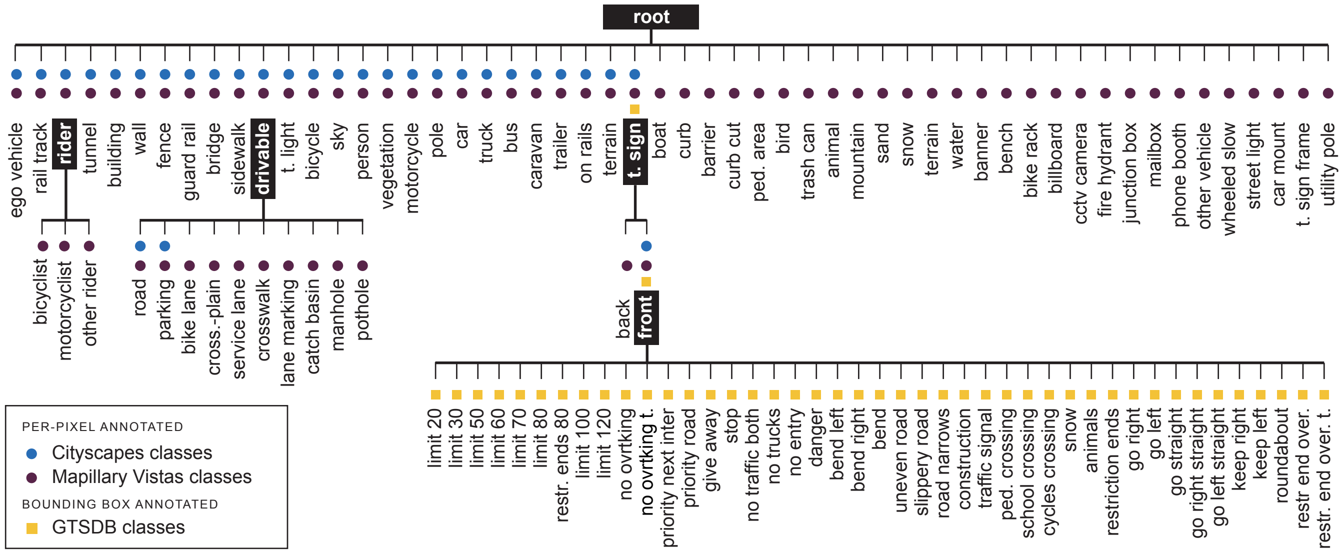

Fig. 2. Three-level semantic label hierarchy combining 108 classes from Cityscapes, Mapillary Vistas and GTSDB dataset. Classes marked in black correspond to the L1, L2, and L3 classifiers of Fig. 1.

The semantic hierarchy of the labels induces a corresponding hierarchy of classifiers. Each classifier classifies the children labels of a node and the whole tree of classifiers is trained, in an end-to-end, fully convolutional manner, over a shared feature representation.

B. *Convolutional network architecture*

The proposed network architecture (see Fig. 1 for an example) consists of a fully convolutional feature extractor for computing a dense, shared representation, and a set of classifiers, each corresponding to an inner class node of the semantic hierarchy. Every classifier can be connected with classifiers one level down in the hierarchy, in order to pass its predictions for inference, and annotation type independent training, as described in Sec. III-C, III-D. Each classifier may be preceded by a shallow *adaptation network*, which adapts the common representation, its depth, and receptive field to the needs of the classifier. This gives the network designer the opportunity to select different features dimensions and receptive fields for each of the classifiers. For example, discriminating between *e.g.* traffic signs is easier [12], as less features are needed, compared to high-level discrimination, like road vs. sidewalk and bushes vs. trees [3]. The flexibility of applying a different field-of-views to different classifiers, enables more or less context aggregation, depending on the classifier's object average size: *e.g.* traffic signs appear generally in smaller scales than buildings or cars.

C. *Inference: hierarchical decision rule*

Inference is carried out per-pixel, in a hierarchical manner across the tree of softmax classifiers. Each classifier j computes a per-pixel normalized vector $\boldsymbol{\sigma}^{j,p}$ of class probabilities for its own set of pixels $p \in P^j$ and set of classes $C^j = \{0, 1, ...\}$, and outputs per-pixel decisions $\hat{y}^{j,p} = \text{argmax}_i \, \boldsymbol{\sigma}_i^{j,p}$, where $\hat{y}^{j,p} \in C^j$. The set of pixels P^j, for which every classifier must produce decisions, is generated by its parent according to its own decisions. Every pixel of the input is labeled with the desired level of detail, from the available set of labels $\{\hat{y}^{j,p}\}_{j \in J}$, where J are the classifiers that produced decisions for this specific pixel.

D. *Training: hierarchical classification loss*

The annotations type of many datasets are not compatible with the required per-pixel supervision for semantic segmentation, as outlined in Sec. II. Our proposed method treats incompatible annotations with a unified approach, without the need of external components, as in [6], [7], and with negligible computational load to the system. The flexibility for handling diverse ground truth, is exchanged with the only constraint that classes on the root classifier, should have per-pixel annotated examples. Annotations for any other level can be of any type or even mixed.

We propose a hierarchical classification loss, which separates supervision according to annotations type at the pixel level. Each classifier j is trained on all labeled pixels $P^j = P_1^j + P_2^j$ that correspond to its respective node in the label hierarchy. Pixels P_1^j with per-pixel annotations are trained using the standard one-hot cross-entropy loss. Pixels P_2^j with non-per-pixel annotations are trained with generated per-pixel ground truth using a modified cross-entropy loss. To achieve this, our method uses the online, per-pixel decisions of the parent classifier during training, to refine the pseudo, per-pixel labels. The process is illustrated in Fig. 3. First, non-compatible annotations are converted to per-pixel pseudo ground truth. Then, in every training step, decisions of parent classifiers are intersected with this pseudo ground truth to generate the per-pixel ground truth used for supervision.

Both losses are accumulated per classifier to the so called hierarchical loss:

$$L^j = -\frac{1}{\left|P_1^j\right|} \sum_{p \in P_1^j} \log \boldsymbol{\sigma}_{y^{j,p}}^{j,p} - \frac{1}{\left|P_2^j\right|} \sum_{p \in P_2^j} \log \boldsymbol{\sigma}_{y^{j,p}}^{j,p} \,, \quad (1)$$

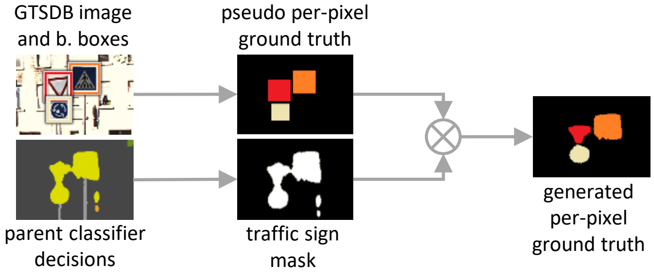
GTSDB image and b. boxes
pseudo per-pixel ground truth
parent classifier decisions
traffic sign mask
generated per-pixel ground truth

Fig. 3. Online procedure during training for generating per-pixel ground truth from bounding box labels.

where $|\cdot|$ is the cardinality of the pixel's set, and $y^{j,p} \in C^j$ selects the element of $\boldsymbol{\sigma}$ that corresponds to the ground truth class of pixel p for classifier j. Finally, losses from all classifiers are collected and weighted with different hyperparameters λ^j to obtain the total objective to minimize:

$$L^{total} = \sum_j \lambda^j \cdot L^j + regularizer \; . \qquad (2)$$

IV. Three-level Label Hierarchy with Cityscapes, Mapillary Vistas and GTSDB

In this section, we outline implementation details, to improve the repeatability of our experiments.

Convolutional network architecture: The network is depicted in Fig. 1. The feature extractor consists of the feature layers of the ResNet-50 architecture [13], followed by an 1x1 convolutional layer (with ReLU and Batch Normalization), to decrease feature dimensions to 256. The stride on the input is reduced from 32 to 8, using dilated convolutions. The representation has depth 256, spatial dimensions 1/8 of the input, and is shared among 5 branches. Each branch has an extra bottleneck module [13] and ends at a softmax classifier, which includes a hybrid upsampling module. We choose the feature dimensions and the field-of-view of the per-classifier adaptation subnetworks to be the same for all branches. After experimenting with different upsampling techniques (fractional strided convolution, bilinear, convolutional), we concluded that the best performance and reduction of artifacts, is obtained by hybrid upsampling, which consists of one 2x2 learnable fractional strided convolutional layer, followed by bilinear upsampling to reach input dimensions.

Implementation details: We use Tensorflow [14] and a Titan X (Pascal architecture) GPU for training and inference. Due to limited memory we set batch size to 4 (Cityscapes:Vistas:GTSDB = 1:2:1) and training dimensions to 512x706 (the average of Vistas images scaled to the smaller Cityscapes dimension). During training images are downscaled, respecting their aspect ratio, and then randomly cropped. The network is trained for 17 Vistas epochs (early stopping) with Stochastic Gradient Descent and momentum of 0.9, L2 weight regularization with decay of 0.00017, initial learning rate 0.01 that is halved three times, and batch normalization and exponential moving averages decay are both set to 0.9. The hyperparameters λ^j of Eq. (2) are chosen to be 1.0, 0.1 and 0.1 for the three levels of the hierarchy respectively. For inference, we currently achieve a frame rate of 17 fps, *i.e.* 58 milliseconds per frame.

V. Evaluation

We conduct the following experiments to evaluate our hierarchical classification approach:

1) **Baselines for flat classification**: Sets the baselines of flat classifiers for single and multiple datasets training.
2) **Hierarchical classification on three heterogeneous datasets**: Demonstrates the benefits of our complete method for combined training on three *heterogeneous* datasets (Cityscapes, GTSDB, Vistas) with disjoint label spaces and different annotation types.
3) **Hierarchical versus flat classification on Cityscapes Extended**: Validates the effectiveness of our hierarchical approach against extremely imbalanced classes, by isolating it on the per-pixel annotated Cityscapes Extended dataset with a two-level label space.

A. Datasets

We summarize the datasets that we use in Table I. Next we describe the extra annotations needed for our experiments. Please note that these annotations are only used for validation purposes and not for training the networks.

1) Labeling Cityscapes with traffic sign classes: We extend the label space of Cityscapes with 43 traffic sign classes of GTSDB. Cityscapes provides only per-pixel traffic sign annotations without differentiating between instances. We design an automatic segmentation algorithm based on the 8-neighborhood distance, for separating connected traffic sign instances in the ground truth traffic sign mask, and a GUI application, which proposes image areas for labeling. We pack original and new annotations under the name Cityscapes Extended. This dataset contains 2778 and 380 traffic signs in the train and validation splits respectively.

2) Annotating GTSDB with per-pixel labels: Only for specific experiments that involve the flat classifier, we converted the GTSDB bounding box annotations to fine per-pixel annotations, using the traffic signs shapes (circle, triangle, hexagon) and inscribing them into their bounding box. This procedure can be problematic with in-plane rotation of traffic signs, but after dataset inspection, we observed that only a negligible amount of in-plane rotations are present.

TABLE I

Dataset statistics. Images contain training and validation splits. In parenthesis the number of evaluated classes are shown.

	Cityscapes (fine)	Cityscapes Extended	Mapillary Vistas	GTSDB
resolution	1024 x 2048		0.5 - 25 MP	800 x 1360
images	2975 / 500		18000 / 2000	600 / 300
annot. type	per-pixel		per-pixel	bound. box
annot. pixels	$1.6 \cdot 10^9$		$156.2 \cdot 10^9$	$0.003 \cdot 10^9$
classes	34 (27)		66 (65)	43 (28)
t. sign classes	-	43 (18)	-	43 (28)
traffic signs	-	3158	-	900

TABLE II

FLAT CLASSIFICATION PERFORMANCE BASELINES ON PER-PIXEL ANNOTATED DATASETS.

	Same dataset			Cross-dataset
Tested on	Cityscapes	Vistas	GTSDB	Cityscapes Extended traffic sign classes
mPA (%)	53.6	36.5	25.4	19.1
mIoU (%)	46.2	29.6	17.2	3.0
Trained on	Cityscapes	Vistas	GTSDB	Cityscapes + GTSDB

B. Metrics and evaluation conventions

We use multi-class mean Pixel Accuracy (mPA) and mean Intersection over Union (mIoU), which are relevant in the context of automated driving, and they represent good local and area criteria, following the definitions given in [15]. For Cityscapes, we report results on 27 classes (19 of the official benchmark + 8 common with Vistas). For the traffic sign classes we evaluate on a subset of the 43 traffic signs that satisfy both conditions: 1) have less than 10^3 pixels in GTSDB train set, and 2) have less than 10^3 pixels on both GTSDB and Cityscapes Extended validation sets. Please note that 10^3 pixels limit is chosen, as it is 2 orders of magnitude smaller than the least represented class in Cityscapes. For Vistas, we report results on the official 65 classes benchmark. Finally, we evaluate the performance of the model every one epoch and we report the average over the last 2 runs.

A new evaluation protocol for fair comparisons is introduced only for the experiments of Sec. V-C, which trains a flat classifier on two datasets. It solves the semantic conflict of the high-level traffic sign class being in the same level with traffic sign subclasses (Sec. II). The decision for a traffic sign pixel is deemed correct: 1) if it is correctly labeled with any traffic sign subclass, or 2) if it is labeled as traffic sign and the second most probable choice is the correct traffic sign subclass. To be clear, we do not use this evaluation scheme for the hierarchical classifier but only for the flat classifier.

C. Baselines for flat classification

In Table II, we set the same- and cross-dataset baselines for the conventional flat classification approach, using the same input dimensions and batch size as described in the implementation details of Sec. IV, in order to be able to fairly compare with the hierarchical results of Table III. In columns 1-3, we train three models on three datasets independently and we provide results for the evaluated classes of Table I. In column 4, we provide cross-dataset results on Cityscapes Extended of combined training on Cityscapes and GTSDB.

For fair comparisons, the models of third and fourth column were trained with the generated per-pixel annotations of the GTSDB dataset (see Sec. V-A.2 for the details). Training on 43 classes of GTSDB does not converge, due to the limited number of training pixels per image, so we included the unlabeled pixels as an extra class, to solve this issue. It can be observed that simultaneous training on Cityscapes and GTSDB, fails to achieve satisfactory cross-dataset results on the traffic sign classes of Cityscapes Extended.

TABLE III

PERFORMANCE OF OUR COMPLETE HIERARCHICAL CLASSIFICATION APPROACH ON 4 DATASETS.

	Same dataset			Cross-dataset
Tested on	Cityscapes classes	Vistas classes	GTSDB classes	Cityscapes Extended traffic sign classes
mPA (%)	66.6	38.9	57.7	29.7
mIoU (%)	57.3	31.9	41.5	8.3
Trained on	Cityscapes + Vistas + GTSDB			

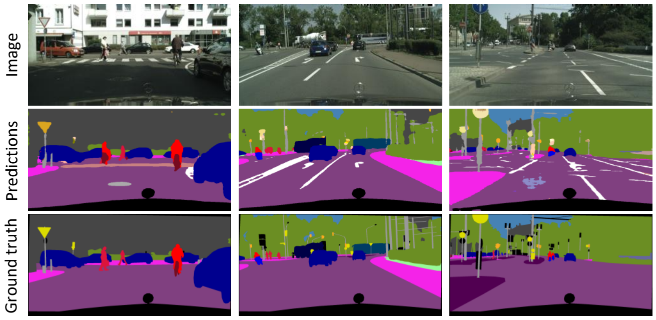

Fig. 4. Cityscapes val split image examples. The network predictions include decisions from L1-L3 levels of the hierarchy. Note that the ground truth includes only one traffic sign superclass (yellow) and no road attribute markings.

D. Hierarchical classification on 3 heterogeneous datasets

This experiment evaluates our complete hierarchical classification approach on three *heterogeneous* datasets (Cityscapes, Mapillary Vistas, and GTSDB).

In Table III, we present evaluation results on the validation splits of the three datasets that the model is trained on (columns 1-3) and results for traffic sign subclasses on Cityscapes Extended (column 4), which was not used during training. In Figures 4, 5, 6 qualitative results are depicted.

By comparing Table II columns 1-3 and Table III columns 1-3, we achieve significant performance increases in mean PA (in the range +2.4% to +32.3%) and IoU (in the range +2.3% to +24.3%) for all datasets. By comparing Table II column 4 and Table III column 4, we also observe an increase in cross-dataset performance on traffic sign subclasses. It is important to note that the model was not trained on any examples from Cityscapes Extended traffic sign classes, and the +10.6% increase in mean PA is solely due to our hierarchical multiple dataset training scheme.

We conclude that hierarchical classification is highly advantageous for combined *heterogeneous* datasets training, when datasets have different classes, different annotation types and in- and between-dataset(s) imbalances.

E. Hierarchical vs flat classification on Cityscapes Extended

In this experiment we evaluate the hierarchical classification method on Cityscapes Extended with per-pixel annotations and a two-level label space. The goal is to isolate our method on a single dataset, in order to show its effectiveness on highly imbalanced datasets against flat classification. We used 512 x 1024 input dimensions and batch size of 2.

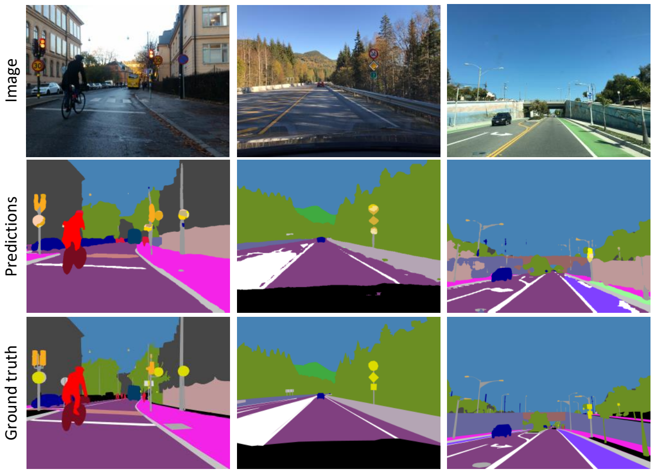

Fig. 5. Mapillary Vistas validation split image examples. The network predictions include decisions from L1-L3 levels of the hierarchy. Note that the ground truth does not include traffic sign subclasses.

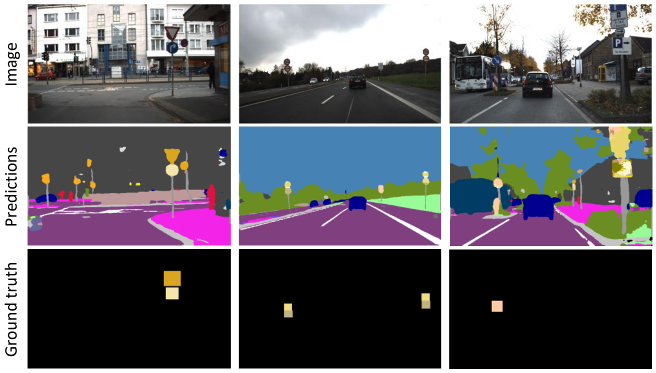

Fig. 6. GTSDB test split image examples. The network predictions include decisions from L1-L3 levels of the hierarchy. Note that the ground truth includes only traffic sign bounding boxes, since rest pixels are unlabeled.

From Table IV, we observe that hierarchical classification significantly increases mPA (+26.0%) and mIoU (+16.1%) for L2 classes (i.e. GTSDB traffic sign subclasses) with respect to the flat classifier, while for L1 classes (i.e. Cityscapes classes) the increase is above +6% for both mPA and IoU.

We conclude that hierarchical classification is robust against class imbalances, even when applied on a single dataset with per-pixel annotations, since it places in each level classes with the same order of examples.

VI. Conclusions and Future Work

In this paper, we considered the challenge of simultaneously training on three *heterogeneous*, but semantically connected datasets, for the problem of per-pixel semantic segmentation. The main motives are to maximally reuse resources (datasets and computation) and eliminate human labeling effort. In order to achieve this, we leverage the semantic relationships between datasets' labels to construct a hierarchy of classifiers and we introduce the respective hierarchical training and inference rules. Our final network can per-pixel segment an input image into 108 classes from 8 high-level street scene categories. The results clearly show the benefit of using our hierarchical classification scheme for multiple heterogeneous dataset training. In future work, we will extend our implementation to include more datasets with more divergent characteristics, to demonstrate the scalability of our approach.

TABLE IV

Flat versus proposed hierarchical classification performance on Cityscapes Extended. (In parenthesis performance for traffic sign L1 class.)

	flat classifier		hierarchical classifiers	
	L1 classes	L2 classes	L1 classes	L2 classes
mPA (%)	69.4 (73.0)	**23.0**	75.6 (74.8)	**49.0**
mIoU (%)	60.4 (65.2)	**12.7**	66.7 (65.7)	**28.8**
trained on	Cityscapes Extended L1 and L2 classes			

References

[1] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," *CoRR*, vol. abs/1704.05519, 2017.

[2] B. Zhao, J. Feng, X. Wu, and S. Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," *International Journal of Automation and Computing*, vol. 14, no. 2, pp. 119–135, Apr 2017.

[3] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy*, 2017, pp. 22–29.

[5] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in *Neural Networks (IJCNN)*. IEEE, 2013.

[6] L. Ye, Z. Liu, and Y. Wang, "Learning Semantic Segmentation with Diverse Supervision," *arXiv preprint arXiv:1802.00509*, Feb. 2018.

[7] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1742–1750.

[8] A. Petrovai, A. D. Costea, and S. Nedevschi, "Semi-automatic image annotation of street scenes," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 448–455.

[9] J. Zhou, Z. Zhou, and L. Zhang, "Hierarchical semantic classification and attribute relations analysis with clothing region detection," in *Advanced Multimedia and Ubiquitous Engineering*. Springer, 2016.

[10] X. Mao, S. Hijazi, R. Casas, P. Kaul, R. Kumar, and C. Rowen, "Hierarchical cnn for traffic sign recognition," in *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 2016, pp. 130–135.

[11] "Full implementation code for training, evaluation and inference, and the extra annotated datasets will be made publicly available at https://github.com/pmeletis/hierarchical-semantic-segmentation."

[12] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 3642–3649.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[15] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.