Train Here, Deploy There: Robust Segmentation in Unseen Domains

Eduardo Romera[1], Luis M. Bergasa[1], Jose M. Alvarez[2] and Mohan Trivedi[3]

***Abstract*— Semantic Segmentation methods play a key role in today's Autonomous Driving research, since they provide a global understanding of the traffic scene for upper-level tasks like navigation. However, main research efforts are being put on enlarging deep architectures to achieve marginal accuracy boosts in existing datasets, forgetting that these algorithms must be deployed in a real vehicle with images that were not seen during training. On the other hand, achieving robustness in any domain is not an easy task, since deep networks are prone to overfitting even with thousands of training images. In this paper, we study in a systematic way what is the gap between the concepts of "accuracy" and "robustness". A comprehensive set of experiments demonstrates the relevance of using data augmentation to yield models that can produce robust semantic segmentation outputs in any domain. Our results suggest that the existing domain gap can be significantly reduced when appropriate augmentation techniques regarding geometry (position and shape) and texture (color and illumination) are applied. In addition, the proposed training process results in better calibrated models, which is of special relevance to assess the robustness of current systems.**

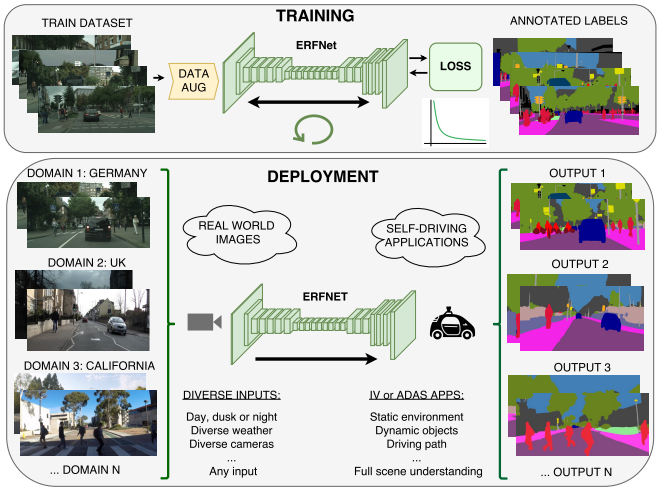


Fig. 1. Overview diagram depicting the proposed end-to-end solution for training and deployment of deep models for robust segmentation. The displayed outputs are real segmentation results produced in our experiments.

I. Introduction

In the last years, the research fields of Computer Vision (CV) and Intelligent Vehicles (IV) have grown together with the aim of solving many of the perception challenges that autonomous vehicles will have in the future. One of the best examples of this alliance is Semantic Segmentation (SS), a vision task that consists of labeling categories in an image at the pixel-level. It has gained high interest in the IV community since it provides a global understanding of the scene at once, allowing to unify several perception tasks that are needed for safe vehicle navigation [1].

Convolutional Neural Networks (CNNs) have recently gained momentum as the best algorithms to perform SS. They have proliferated in the last years due to an incessant increase in affordable computational resources and due to the appearance of large datasets to train these data-hungry methods. In the particular context of autonomous driving, embedded devices have grown more computationally powerful, and large datasets like CamVid [2] and Cityscapes [3] have extremely facilitated the tasks of training and testing deep models for segmentation. However, main efforts on improving segmentation methods have been focused on increasing accuracy while leaving efficiency as a second priority [4][5][6]. In this context, we presented ERFNet, a convolutional architecture that produces SS both accurately and efficiently, delivering a good trade-off that is convenient for IV applications like autonomous driving [7][8]. However, what remains practically unexplored is robustness to unseen driving scenarios. After all, CNNs are trained from a limited set of data and there is no guarantee that the knowledge learned (from a dataset) is transferred properly to any domain. For any deep model trained with limited data, there is a question that remains unanswered: How well does it perform in unseen environments?

In this paper, we aim to solve this question, for the specific task of semantic segmentation, by analyzing what is the gap between the concepts of "accuracy" and "robustness". We study what specific measures can be addressed to improve CNN's robustness to make them perform more accurately in environments/domains that were unseen during training. More precisely, we experiment with our publicly available architecture ERFNet [8], but we study these problems in a general way that is applicable to any other deep architecture. Our comprehensive set of experiments with datasets from multiple domains demonstrates that data augmentation plays an essential role in achieving robustness in deployed end-to-end segmentation architectures.

*This work has been funded in part from the Spanish MINECO/FEDER through the SmartElderlyCar project (TRA2015-70501-C2-1-R) and from the RoboCity2030-III-CM project (S2013/MIT-2748), funded by Programas de actividades I+D (CAM) and cofunded by EU Structural Funds. The authors also thank NVIDIA for generous hardware donations.

[1]Eduardo Romera and Luis M. Bergasa are with the Electronics Department, University of Alcalá (UAH), Spain eduardo.romera@edu.uah.es, luism.bergasa@uah.es

[2]José M. Alvarez is with CSIRO and the Australian National University (ANU), Australia Jose.Alvarezlopez@data61.csiro.au

[3]Mohan Trivedi is with the Laboratory for Intelligent and Safe Automobiles, University of California San Diego (UCSD), USA mtrivedi@ucsd.edu

II. Related Works

SS advanced rapidly since Long et al. [9] proposed to adapt known CNNs to produce pixel-wise classification outputs by using convolutions as the last layers. These Fully Convolutional Networks (FCNs) achieved surprising results in segmentation datasets while also being a convenient end-to-end solution. However, SS does not only require a classification output per image, but one per pixel. In practice, this involves that the output produced by pretrained CNNs (i.e. transferred from the classification task) is coarse, since these features have not been specifically trained to learn pixel localization. To address this issue, there have been several works that have tried to enhance the way that CNNs learn about context. The work in [4] (Deeplab) proposed to add Conditional Random Fields as a post-processing step to refine the coarse convolutional output. SegNet [5] proposed to attach a full classification network with max-unpooling layers as a decoder that produces end-to-end pixel-wise classification from encoded features. The work in [6] proposed to virtually "dilate" the convolutional kernels to make them gather more context information.

All these works contributed to achieve substantial accuracy improvements in well-known segmentation benchmarks. However, their main efforts were focused on accuracy by assuming that efficiency was not a top priority. On the other hand, other networks like ENet [10] were proposed as an efficient alternative to perform fast semantic segmentation in real-time. One hundred layers tiramisu [11] is also an efficient extension of DenseNet to perform SS. However, these works sacrify some of the accuracy earned by more complex architectures in order to remain efficient. In a previous work, we proposed ERFNet [7][8], which aimed to maximize the trade-off between accuracy/efficiency and make CNN-based segmentation suitable for IV applications in current embedded hardware platforms.

Despite these recent advances, it is still unclear how these networks generalize in unseen domains in everyday driving situations. For now, segmentation networks must learn from labeled data in a supervised way to achieve top accuracy. Datasets like CamVid [2] and Cityscapes [3] have hundreds of images, but even their diversity does not guarantee top performance in any unseen scenario in the real world. In the field of domain adaptation, Ros et al. [12] proposed an unsupervised color transformation approach to adapt the images of the training domain to other illumination conditions (e.g. transfer between daytime and dusk). Other works have addressed the lack of samples by generating synthetic data (e.g. SYNTHIA [13]). However, transferring learned features from the virtual domain to the real one is not an easy task. Even though simulators are constantly improving and they currently produce more realistic images than ever, deep models trained specifically with synthetic data still don't perform well in real domains. We argue that simulators are still producing noise that is not correlated to the real domain and the capacity of current deep models is making them overfit that noise. In this context, where data annotation is extremely time consuming and synthetic data isn't helping, the deep learning community is shifting their efforts to unsupervised models (e.g. GANs) to avoid this high dependence on annotated data. However, we argue that there are existing measures that can be applied now in order to produce robust segmentation models that can be deployed in any domain and can be used to address current IV challenges.

III. Method

Deep architectures have a high dependence on the data that is used for training, since the features that are learned by a CNN rely entirely on the images that are fed in this process. Therefore, data diversity plays an essential role in achieving models that are more general, due to the wide variety of patterns that CNNs need to learn to be able to discriminate well between categories. In this section, we describe a wide range of methods that aim at augmenting a limited set of data to improve robustness. Most of these techniques are known and some of them are widely used as a common practice while training CNNs. Among these techniques, some have an effect on the geometry of the categories (i.e. position and shape) and others have an effect in the texture (i.e. illumination and color). Both, geometry and texture, affect how the CNN learns patterns from the training images in order to produce the semantic segmentation output. Therefore, it is essential to augment both in order to improve the network performance in unseen domains.

Geometric augmentations:

1) **Horizontal flip**: mirroring the image horizontally helps to add invariance to orientation (e.g. a pedestrian can appear with diverse orientations). Vertical flipping is not recommended since the vertical appearance of objects adds important consistency in the scene (e.g. the network knows what sky is due to its position).
2) **Translation**: Moving the image prevents the CNN from seeing always the same position of the training images, so it doesn't always generate the same activations from the very first layer (shift invariance). In our experiments, we use random translation of 0-2 pixels since the first layer of ERFNet is a 3x3 convolution.
3) **Scaling and Cropping**: random resizing images helps the model see diverse scales of each object and improves network invariance to diverse image resolutions. We perform random scaling uniformly between 0.5 and 1.0 times the original size. We combine it with randomly cropped regions of the image to keep the same resolution in the training batch. Please note that crops also add shift invariance like Translation.
4) **Aspect ratio**: Rescaling the image in one dimension (width or height) helps adding invariance against diverse aspect ratios (e.g. 4:3, 16:9) that can be specific to each camera. In our experiments, we rescale the width between (0.7 and 1.0) times uniformly.
5) **Rotation**: Rotating a small random degree to the image adds invariance against objects that might appear with slight angle variations in the scene. We rotate the

whole image a random amount of radians following a Gaussian distribution with mean 0 and 0.05 variance.

Texture augmentations:

1) **Brightness**: how clear the objects appear in the image depends on the scene illumination and camera sensitivity. Adding virtual alterations to the input images by randomly increasing or decreasing the image brightness improves network's illumination invariance. In our experiments, we alter brightness following an uniform distribution between 0 and 0.4.
2) **Contrast**: separation between the darkest and brightest areas of the image. Increasing this range with random augmentations helps adding invariance against shadows and generally improves network's performance in low lighting conditions. We augment contrast uniformly between 0 and 0.4 w.r.t. grayscale mean.
3) **Saturation**: depth or intensity of the color. The lower the saturation, the less intense are the colors. Augmenting this parameter adds invariance to different camera sensitivities to capture color. We augment saturation by altering color channels uniformly between 0 and 0.4.
4) **Color Jitter**: Adding small random noise to each RGB pixel helps obtaining invariance against some camera distortions. We add Gaussian-distributed jitter to each channel's pixel with 0 mean and 0.05 variance.
5) **Salt & Pepper**: similar to color jitter, but saturating specific pixels to black or white with random probability. Old cameras had this kind of distortion in the past but this is uncommon in recent ones. In our tests we tried saturating 2.5%, 5% and 10% pixels, and none helped improving accuracy in recent datasets.

IV. Experiments

Accuracy is normally measured in specific datasets and it is often taken as a measure of how robustly will perform the model in any scenario. However, datasets are normally recorded on specific conditions and they do not represent the diversity of the real world. On the other hand, evaluating robustness numerically is challenging due to the lack of labeled data in different domains. In this paper, we experiment with well-known labeled datasets Cityscapes and CamVid for the main quantitative experiments (ablation studies and comparison with other networks) and finally we show challenging qualitative examples in these datasets and additional data captured in California in diverse conditions. Additionally, we look into the concept of network calibration as a metric to measure robustness.

A. Experimental setup

Cityscapes [3] contains 2975 images for training, 500 for validation and 1525 for testing (not publicly available). CamVid [2] has 701 images in total, split into 367 images for training, 101 for validation and 233 for testing. These images come from 4 sequences, where one was recorded at dusk (124) and the rest were recorded at daylight (577). In order to evaluate the effect of each augmentation technique, we train all models in Cityscapes train set with 19 classes and then test in other domains. Since the categories of CamVid are not easily compatible with the 19 used for training, in CamVid we adapt the main 11 classes (common ones used in the literature) to the closest one in Cityscapes and set the rest to unlabeled (black). About the CNN training setup, we train all models in the same conditions using Adam optimization with an initial Learning Rate (LR) of 1e-4 and Weight Decay (WD) of 2e-4, decreasing LR exponentially until cross-entropy loss converges. For more details about optimal training setup or architecture details please refer to ERFNet papers [7][8]. All numerical results are shown in the widely used "Intersection over Union" ($IoU = \frac{TP}{TP+FP+FN}$).

TABLE I

Analysis on the effect of diverse data augmentations. Models are trained in Cityscapes train set, results are in Cityscapes Validation set (500) and CamVid full set (701).

	Random augments	Cityscapes	CamVid
-	0. Baseline	69.2	41.9
Geometry	1. Horizontal flip	**71.0**	**52.6**
	2. Translation	70.9	43.1
	3. Scale & Crop	70.3	45.3
	4. Aspect Ratio	69.5	46.3
	5. Rotation	70.4	45.9
Texture	1. Brightness	68.4	**59.4**
	2. Contrast	69.0	56.6
	3. Saturation	**69.5**	52.4
	4. Color Jitter	69.3	49.8
	5. Salt & Pepper	67.1	37.8
Combinations	Geometry 1+2	70.0	45.7
	Texture 1+2	68.9	65.8
	Geometry-1 + Texture-1	69.4	52.2
	Top-4 Geometry (1,3,4,5)	69.9	46.4
	Top-4 Texture (1,2,3,4)	69.7	65.9
	All combined (T4-G + T4-T)	**71.2**	**71.5**

B. Data Augmentations

In this experiment, we analyze the effect of each specific data augmentation technique in a systematic way. Results are shown in Table I. All listed models are trained in the same conditions (in Cityscapes Train set) and evaluated in the Cityscapes Val set (500 images) and in the full CamVid dataset (all 701 images).

The results show that there are specific augmentations that have a very high impact in improving the result in CamVid domain while others have a very slight effect. For instance, horizontal flip implies a high boost in IoU (52.6%) compared to the rest of geometric augmentations (43-46%). In the case of texture, all augmentations except Salt&Pepper have a very high effect in boosting CamVid accuracy with respect to the baseline. In the case of brightness augmentation, it almost boosts the IoU to 60%, while contrast, saturation and jitter also reach 50-56%. Compared to Cityscapes, CamVid sequences look much darker in general, so it makes sense that illumination augmentations produce a larger improvement compared to geometric ones. On the other hand, geometric augments have a slight effect in improving result in Cityscapes Val set while texture transforms even deteriorate it in some cases. This makes sense since Train and Val sets in

TABLE II

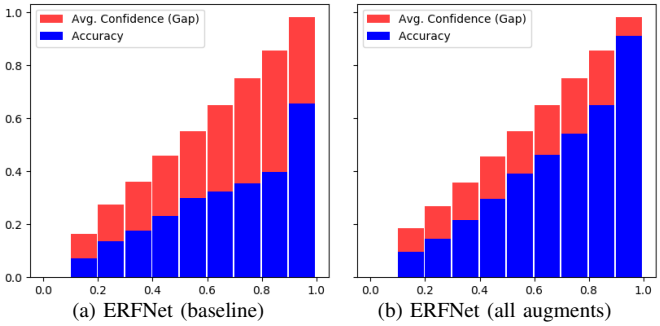
RESULTS IN CITYSCAPES AND CAMVID SUBSETS FOR THE MAIN MODELS TRAINED IN TABLE I. ALL RESULTS ARE IN IOU. RESULTS IN TRAIN SET ARE SHOWN TO GIVE AN INTUITION ABOUT OVERFITTING.

Model	CITYSCAPES		CAMVID			
	Train (2975)	Val (500)	Dusk 01TP (124)	Day 06R0 (101)	Day 16E5 (305)	Day 05VD (170)
Baseline	**85.2**	69.2	13.5	35.6	49.3	51.9
Top4 Geom.	77.4	69.9	10.7	40.5	55.1	57.9
Top4 Texture	84.9	69.7	47.3	56.6	70.4	68.4
All (T+G)	78.3	**71.2**	**61.9**	**58.3**	**74.6**	**70.3**

Cityscapes have similar lighting conditions, so augmenting data with texture transforms does not help much while augmenting the train set with geometric transforms helps the CNN see additional patterns (hence reducing overfitting in the train set and slightly boosting result in val set).

On the other hand, the experiments with combined augments do not confirm the intuition that adding all transformations together by "brute force" always improves result in all domains. For example, adding two geometric transforms (hflip+translation) does not boost CamVid as much as H-flip (52%) but results in a 45.7% IoU. On the other hand, combining the two top texture augmentations (brightness+contrast) does achieve a very high result in CamVid (65.8%), almost like combining the top 4 texture augments (65.9%). These results are reasonable considering that augmentations are in fact virtual transformations of the real images, so adding many augments up may make the model train with too many images that are not similar to the real domain that the CNN will see in deployment (hence reducing the result). In order to train a model that behaved well in both domains (Cityscapes and CamVid), we had to reduce the variance of each augmentation a bit and train significantly more epochs until convergence. The result is a model that achieves very high accuracy in CamVid full set (71.5% IoU) without having seen any image from that domain.

Table II summarizes additional results for the subsets of each dataset: Cityscapes Train and Val sets, and the four CamVid sequences. The results in Cityscapes subsets give insight on how augmenting data with the proposed techniques highly prevents overfitting and helps achieving robust models for deployment. Please note that the IoU in the training set slightly decreases as the IoU in all other unseen data highly increases. The results in CamVid confirm the analysis that the texture augmentations have a greater effect in CamVid data because illumination conditions are very different compared to Cityscapes domain. Please note the high boost in the Dusk sequence (01TP) for the texture-augmented model compared to the baseline and geometry transforms. The result for the "All-augments" model achieves very high accuracy in all sequences while also keeping a high result in Cityscapes domain.



(a) ERFNet (baseline) (b) ERFNet (all augments)

Fig. 2. Reliability diagrams measured in Cityscapes Val set for ERFNet trained with diverse amounts of data-augmentation. Output predictions are divided in uniform bins and the "gaps" reflect the distribution shift of network predictions. Larger gaps (a) reflect that the network is less calibrated than in the case of smaller gaps (b).

C. Network calibration through data augmentation

The network output (class probability) is usually taken as a confidence measure of how sure is the model regarding its prediction. However, if the network is not well "calibrated" (e.g. it has overfitted the train data), it may be overly optimistic about its predictions (e.g. forwarding a very high probability for a class even when it is wrong). To solve this issue, in [14] the authors propose to adapt network output with temperature scaling as a post-processing step. Our experiments with data augmentations reflect that adding variety to the train samples via augmentation does already achieve the desired effect of calibrating the network. In Fig. 2, we show two reliability diagrams (calculated in Cityscapes Val set) which compare the baseline model (a) with the model trained with all augmentations (b). The output predictions are grouped into bins (ranges of probabilities), and in each one we reflect the correctly predicted samples (true accuracy) compared to the average confidence of the network. Larger gaps as in (a) mean that the network is less calibrated compared to the smaller gaps in (b). For a perfectly calibrated model the diagram should look like the identity function. In practice, larger gaps indicate a model predisposition to abrupt output probabilities, which is dangerous in IV applications since the network is very "sure" about accuracy of its segmented output when it shouldn't be.

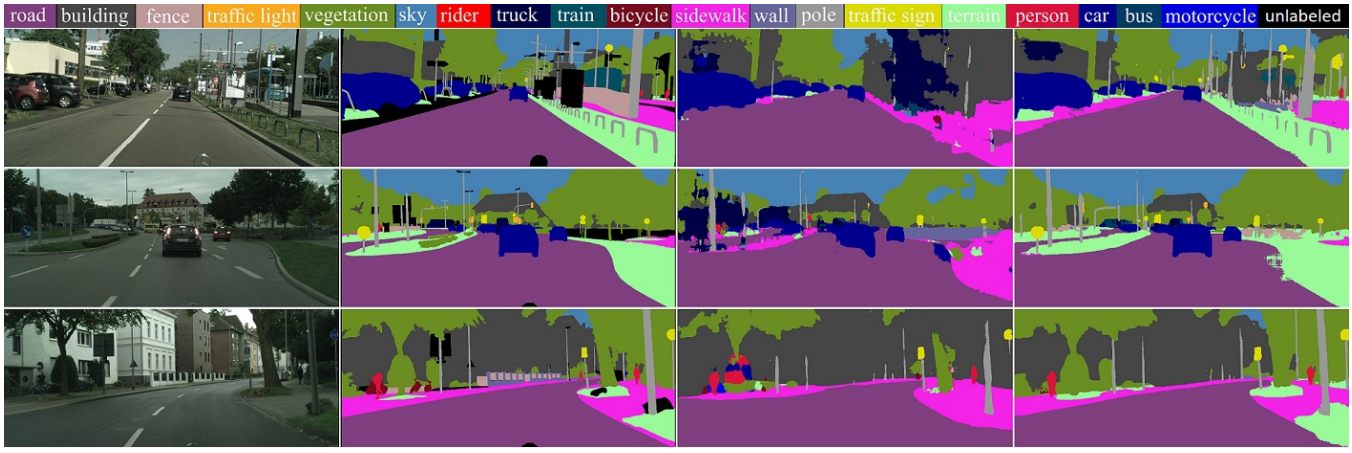
D. Comparison with other networks

In Table III, results are displayed in CamVid Test set and compared with other State-of-the-Art networks. We show results for the 11 main classes like in their papers. All results are in IoU. The percentage of correct pixels per image, or Global accuracy ($Acc = \frac{TP}{TP+FP}$), is also shown for an easier comparison with previous works. Additionally, we have trained ERFNet in CamVid data in the same conditions as the other networks for comparison reasons. The results confirm that using only one domain (Cityscapes data) with a wide range of augmentations reduces the need to train in the specific domain to achieve high accuracy. As shown, the results for our model are similar (or even higher) to the top models in all specific classes. In general, the IoU result for ERFNet with augmentations (68.6%) is higher than all other models, even compared to ERFNet trained in CamVid data.

TABLE III

RESULTS FOR STATE-OF-THE-ART MODELS IN CAMVID TEST SET COMPARED TO OUR MODEL TRAINED USING ONLY CITYSCAPES DATA.

Model	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Cyclist	Mean IoU	Global Acc
SegNet [5]	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	46.4	62.5
ENet [10]	Per-class IoU values are not available in [10]											51.3	68.3
FCN8 [9]	77.8	71.0	88.7	76.1	32.7	91.2	41.7	24.4	19.9	72.7	31.0	57.0	88.0
Deeplab-LFOV [4]	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6	-
Dilation8 [6]	84.0	77.2	91.3	**85.6**	**49.9**	92.5	59.1	37.6	16.9	76.0	57.2	66.1	88.3
FC-DenseNet103 [11]	83.0	**77.3**	**93.0**	77.3	43.9	**94.5**	59.6	37.1	37.8	82.2	50.5	66.9	91.5
ERFNet (CamVid-trained)	80.5	76.8	92.6	83.8	16.0	92.9	65.2	45.9	**46.2**	**87.8**	**63.2**	68.3	**94.3**
ERFNet (Cityscapes+augments)	**85.4**	72.2	83.8	82.0	41.1	93.7	**66.6**	**54.7**	44.6	85.5	48.5	**68.9**	91.2



(a) Input image (b) Ground truth (c) Baseline (no augments) (d) All-augments model

Fig. 3. Examples in Cityscapes Validation Set (500) for a model trained in Cityscapes train set. Each row corresponds to a challenging image in diverse cities (Frankfurt, Lindau and Münster). The color leyend for the 19 cityscapes classes (+ unlabeled) has been added on top for visualization reasons.

E. Qualitative results

For an easier evaluation of how the proposed techniques improve robustness, we display diverse segmentation results in challenging frames of multiple datasets. In Fig. 3, we show results for Cityscapes Validation set (one image per city: Frankfurt, Lindau and Münster). In Fig. 4, results are shown for CamVid dataset (one image row per sequence). Both datasets have pixel-annotated labels available. For additional examples, we have tested our models in an additional domain: California. In Fig. 5, we show results for data recorded in the University of California San Diego (LISA dataset [15]). Diverse results with different illumination conditions (cloudy vs. sunny) and different cameras have been combined in the figure. In summary, it can be seen in all qualitative examples that data-augmentation has an extremely positive effect in improving robustness in all kinds of domains and camera conditions.

V. CONCLUSIONS

In this paper, we have analyzed techniques to be applied to existing deep networks in order to improve their robustness when deployed in any domain. After training models with diverse combinations of data augmentation methods, it has been demonstrated both numerically and qualitatively that these models are ready to produce accurate segmentation in many domains (regardless of place conditions or camera quality). Our systematic and comprehensive set of experiments demonstrates that robustness to unseen domains is reachable with existing techniques that can be applied to any data-driven architecture.

REFERENCES

[1] E. Romera, L. M. Bergasa, and R. Arroyo, "Can we unify monocular detectors for autonomous driving by using the pixel-wise semantic segmentation of cnns?" *arXiv preprint arXiv:1607.00971*, 2016.

[2] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *ECCV*, 2008, pp. 44–57.

[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 *IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2016, pp. 3213–3223.

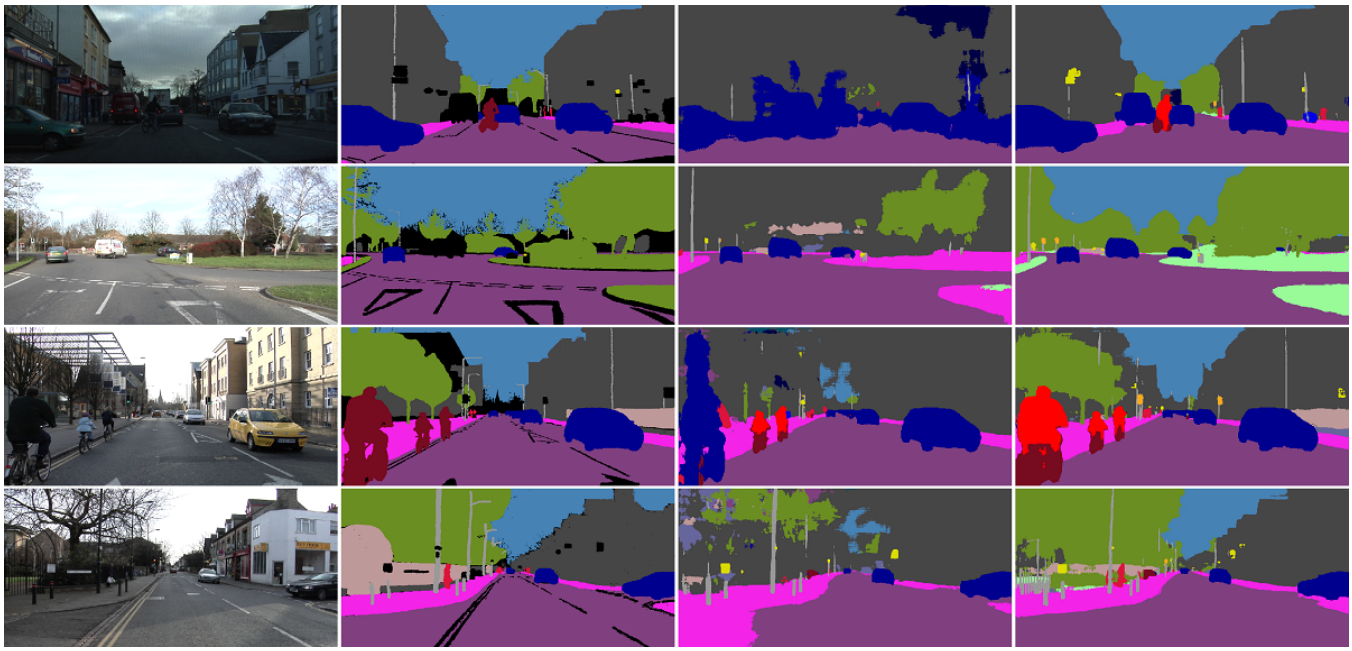
[4] C. Liang-Chieh, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," in *International Conference on Learning Representations*, 2015.

[5] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv:1505.07293*, 2015.

[6] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

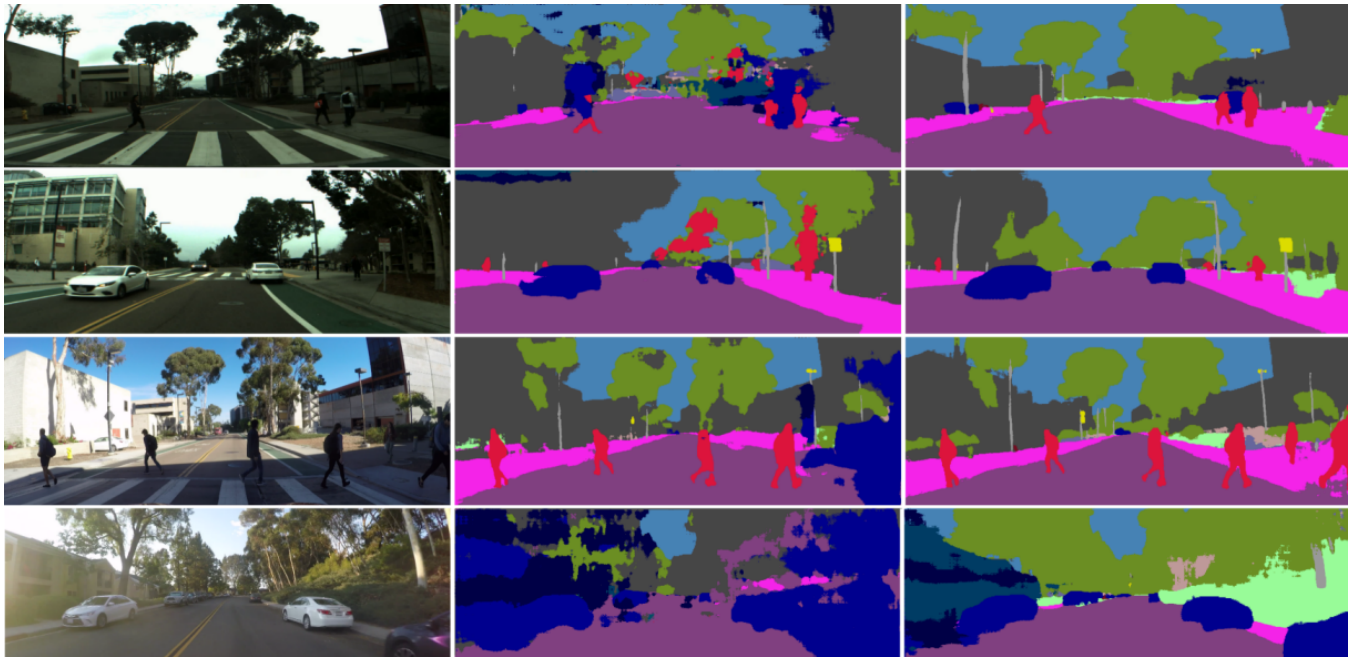
[7] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "Efficient convnet for real-time semantic segmentation," in *IEEE Intelligent Vehicles Symp. (IV)*, 2017, pp. 1789–1794.

[8] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2018.



(a) Input image (b) Ground truth (c) Baseline (no augments) (d) All-augments model

Fig. 4. Examples in CamVid sequences for models trained uniquely with Cityscapes data. Each row corresponds to a challenging image in each of the 4 CamVid sequences. Please note that the CamVid ground truth is only colored for its 11 main classes and the trained models output 19 classes. For example, in CamVid labels, Bicycle and rider are an unique class (dark red) while in Cityscapes they are labeled as two classes (rider and bicycle).



(a) Input image (b) Baseline (no augments) (c) All-augments model

Fig. 5. Examples in LISA data [15], recorded in the University of California San Diego with diverse cameras and illumination conditions.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2015, pp. 3431–3440.

[10] 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.

[11] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," *arXiv preprint arXiv:1611.09326*, 2016.

[12] G. Ros and J. M. Alvarez, "Unsupervised image transformation for outdoor semantic labelling," in *IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 537–542.

[13] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.

[14] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *arXiv preprint arXiv:1706.04599*, 2017.

[15] A. Rangesh, K. Yuen, R. K. Satzoda, R. N. Rajaram, P. Gunaratne, and M. M. Trivedi, "A multimodal, full-surround vehicular testbed for naturalistic studies and benchmarking: Design, calibration and deployment," *arXiv preprint arXiv:1709.07502*, 2017.