2017 IEEE International Conference on Robotics and Automation (ICRA)
Singapore, May 29 - June 3, 2017

AdapNet: Adaptive Semantic Segmentation in Adverse Environmental Conditions

Abhinav Valada[†] Johan Vertens[†] Ankit Dhall[‡] Wolfram Burgard[†]

***Abstract*— Robust scene understanding of outdoor environments using passive optical sensors is a onerous and essential task for autonomous navigation. The problem is heavily characterized by changing environmental conditions throughout the day and across seasons. Robots should be equipped with models that are impervious to these factors in order to be operable and more importantly to ensure safety in the real-world. In this paper, we propose a novel semantic segmentation architecture and the convoluted mixture of deep experts (CMoDE) fusion technique that enables a multi-stream deep neural network to learn features from complementary modalities and spectra, each of which are specialized in a subset of the input space. Our model adaptively weighs class-specific features of expert networks based on the scene condition and further learns fused representations to yield robust segmentation. We present results from experimentation on three publicly available datasets that contain diverse conditions including rain, summer, winter, dusk, fall, night and sunset, and show that our approach exceeds the state-of-the-art. In addition, we evaluate the performance of autonomously traversing several kilometres of a forested environment using only the segmentation for perception.**

I. Introduction

Over the past years, Deep Convolutional Neural Network (DCNN) approaches have achieved impressive results in various visual perception problems including object classification [8], [21], [22], detection [6], [18] and scene parsing [1], [15], [17]. DCNNs have also been used for end-to-end learning of robotic tasks such as detecting robotic grasps [13] and autonomous driving [2]. However, beyond standard benchmarks and new end-to-end learning applications, they have yet to become the go-to-solution for outdoor robotic perception. This can be attributed to two primary impediments: perception in outdoor environments is inherently more challenging due to frequent appearance changes that take place throughout the day and across seasons, and secondly, most existing datasets do not encompass these appearance changes, making techniques that are benchmarked on them to perform poorly in the real-world.

Robust scene understanding is a prerequisite for autonomous navigation. A robot without a perception system capable of handling such large visual changes can very quickly jeopardize its operation and even imperil the people around. Recent work [16], [24] has shown promising results in fusing features from complementary modalities and spectra that are resilient to commonly observed perceptual variations. In this paper, we address the problem of robust semantic segmentation using our proposed AdapNet architecture which incorporates the Convoluted Mixture of Deep Experts (CMoDE) model for dynamically fusing multiple spectra and modalities. Our DCNN consists of three components: experts that map modalities or spectra to segmentation outputs, the CMoDE that adaptively weights class-specific features of expert networks using the learned probability distributions and the fusion segment that further learns complementary fused kernels. We evaluate the performance on publicly available datasets that contain diverse environments such as highways, cities, towns and forests, and in adverse environmental conditions including rain, snow, glare, low-lighting, as well as seasonal appearance changes during summer, winter, dusk, fall, night and sunset. We show that our CMoDE framework outperforms segmentation using current state-of-the-art fusion techniques in the aforementioned scenarios.

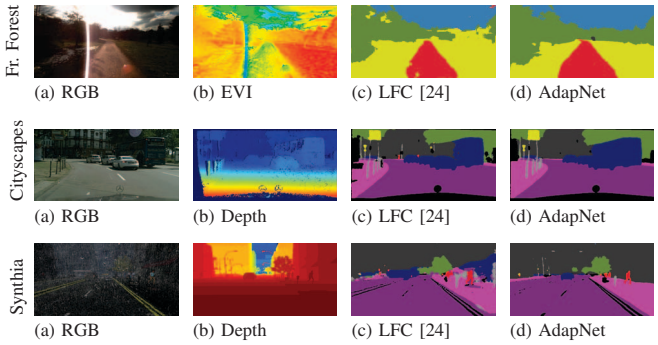

Fig. 1. Qualitative comparison of the segmentation results obtained using AdapNet with LFC [24] and AdapNet with CMoDE on different datasets that contain adverse conditions: performance in a forest scene in the presence of glare directly on the optics and snow (first row), street scene in the presence of motion blur (second row) and with rain (third row).

Overall, this paper makes several contributions. First, we present a new end-to-end semantic segmentation architecture based on the residual learning framework and dilated convolutions. Second, we benchmark the performance of our expert network in comparison to state of the art end-to-end approaches. Third, we introduce the Convoluted Mixture of Deep Experts (CMoDE) fusion scheme for learning robust kernels from complementary modalities and spectra. Fourth, we comprehensively evaluate the performance on three different publicly available datasets that contain diverse scenes in adverse environmental conditions. Finally, we present results from autonomous navigation experiments in a forested environment using the resulting segmentation for perception.

II. Related Work

In recent years, DCNN approaches have achieved unprecedented performance on various semantic segmentation benchmarks. This surge in performance is primarily fuelled by the introduction of Fully Convolutional Networks

[†]University of Freiburg, Germany. [‡]VIT University, India. This work has been supported by Samsung Electronics Co., Ltd. under the GRO program.

(FCNs) [15]. The general structure of such networks consists of an encoder or contraction segment and a decoder or expansion segment. The contraction segment, typically a classification network with inner-product layers substituted with convolutions, maps the input to a low resolution representation, while the expansion maps the low resolution feature maps to upsampled segmentation output. Most approaches [15], [17], [1], [14] utilize the VGG16 [22] for the contraction segment. Long *et al.,* uses the pooling layers from the contraction segment to refine the segmentation in the expansion segment. Oliveira *et al.* [17] proposed improvements that reduce the number of parameters and additional refinement stages that improve the resolution of segmentation. Unlike FCNs which fuse features from different resolutions, Badrinarayanan *et al.,* [1] proposed an approach that takes downsampled features from VGG16 and upsamples them in a decoder segment that uses pooling indices from the decoder segment. In contrast to these existing approaches, we present a novel semantic segmentation architecture that builds upon deep residual networks [8] and dialted convolutions [25], that enable our model to learn very deep representations while aggregating multiscale contextual information. The components of our architecture are elaborated in Sec. III-A.

As perception using unimodal images is excessively sensitive to appearance variations caused by changing environmental conditions, there has been an advent of using alternate modalities to refine the output. In addition to providing the DCNN a richer representation of the scene, using complementary modalities enables the network to learn features that are generalizable across varying conditions. In [9], the authors introduce a multi-spectral dataset containing RGB and thermal images for detecting pedestrians in non-ideal conditions. They also introduce multi-spectral aggregated channel features to handle such data. Eitel *et al.,* [6] use a two-stream DCNN trained on RGB-D data, for object recognition. The outputs of the two streams are concatenated in the end and passed through a *softmax* layer to yield a combined prediction. In our previous work [24], we use a similar approach for segmentation, but in addition to combining the feature maps from multiple streams, their model learns fused features using a stack of convolution and pooling layers. They experiment with multiple spectra and modalities and show that their Late-fused Convolution approach achieves state-of-the-art performance.

Another approach to fusing multiple specialized networks is related to the Mixture of Experts (MoE) model. Hinton *et al.,* [10] presented the classical MoE model with E-experts and a supporting gating network, where the experts map the input X to the output y, whereas the gating network produces a probability distribution over the experts. Hwang *et al.* [5], extend the concept of a MoE by employing DCNNs as experts to classify MNIST and monophone speech data. Unlike the approach that we introduce in this paper, they use the same input X for each expert which is also used to train the gating network. This paper highlights the fact that using a mixture with deep networks increases the number of trainable parameters without significantly increasing the computational burden. Mees *et al.,* [16] employ a mixture of deep experts to detect people in varying illumination using RGB and depth experts. They demonstrate how each expert overcomes the shortcomings of the other, for instance, the depth network yields more reliable detections in a dark corridor than the RGB expert. Their gating network uses inner product layers to produce two scalars, which are then used to take a weighted average over the outputs.

Both the categories of approaches mentioned above have their own benefits and drawbacks. The MoE approach gives a network the ability to adaptively weight experts based on the input, whereas the late-fusion approach enables the network to learn complementary fused kernels. In this work, we propose CMoDE, that exploits the benefits of both these contrasting techniques. Our proposed fusion scheme empowers the network with the ability to choose class-specific features from expert networks based on the current scene representation, followed by learning deeper representations from the mixture of kernels from the experts. More specifically, our model first weighs experts class-wise based on the learned probability distribution and then learns to leverage complementary features from experts that are robust to the observed environmental disturbances. We show that the CMoDE approach not only exceeds the performance of existing fusion techniques, but more importantly provides an accurate representation of scene in adverse conditions.

III. Technical Approach

In this section, we first describe our expert architecture for segmenting unimodal images. Subsequently, we present our CMoDE fusion scheme for learning complementary fused kernels from multiple expert networks. We then detail the network training procedure that we employ, followed by a description of datasets on which we benchmark.

A. Expert Network

1) AdapNet Architecture: Our architecture follows the general principle of having a contractive and an expansive segment, similar to the FCN architectures described in Sec.II. In contrast to the previous approaches, for the contractive segment, we adapt the recently proposed ResNet-50 [8], which has demonstrated impressive results in the ImageNet classification challenge and has recently been adapted for disparity estimation [12]. The ResNet architecture includes batch normalization and layers that can skip convolutions. This allows the design of much deeper networks without facing degradation of the gradient and therefore leads to very large receptive fields with often highly discriminative features. The output of this contractive segment is 32-times downsampled with respect to the input. We then upsample the output of this contractive segment using deconvolutions and perform refinement similar to [15] by fusing high resolution feature maps from the contractive segment . We denote this base architecture as ResNet Upconv in our experiments described in Sec IV-C. The performance of ResNet Upconv is almost as high as FCNs and in addition of having significantly lesser parameters and much faster forward-pass time's, which are critical characteristics for deploying models on embedded GPUs used in robotic perception applications.

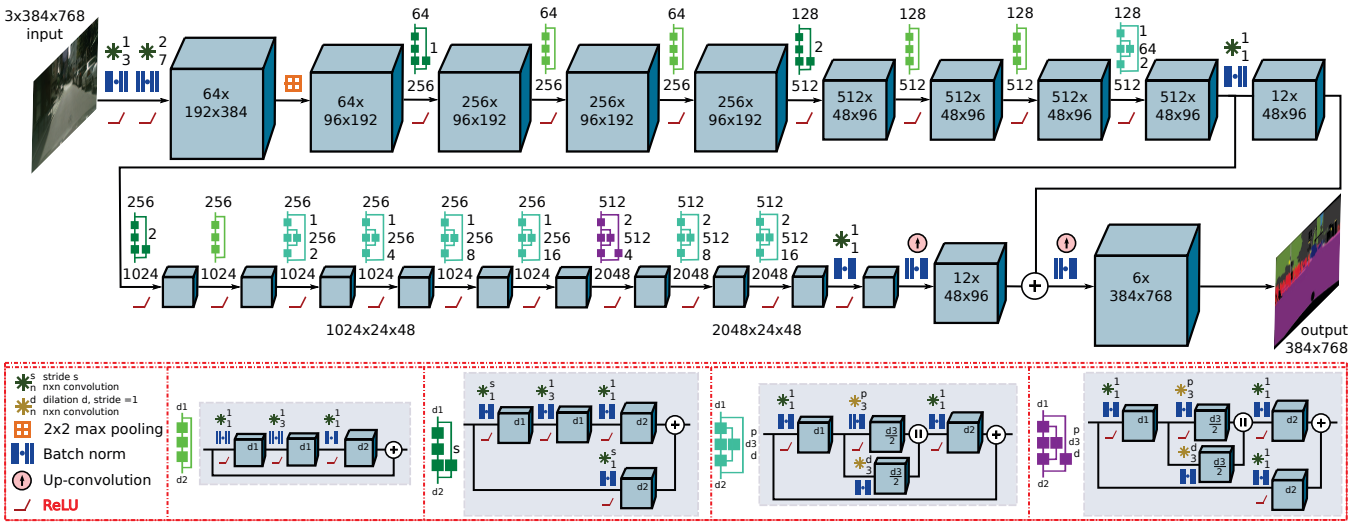

Fig. 2. Depiction of the proposed AdapNet architecture. We additionally add a convolution at the beginning of the network and change the last convolution from a stride of two to one. We also change the convolutions that follow to an atrous convolution with $r = 2$. The lower left two blocks in the legend (enclosed in red) show the original ResNet blocks, while the lower right two blocks show the corresponding proposed multiscale blocks.

While training, the network converges in about 12 hours on a NVIDIA TITAN X, in comparison to FCNs which take about three days.

In order to improve the performance of this base ResNet Upconv model, we propose the following:

a) Front Convolution (FC): In the ResNet architecture, the resolution of the feature maps drops down to a fourth of the input resolution after passing through the first 3 layers. On one hand, this allows for context aggregation and speed-up due to smaller feature maps, but on the other hand, this restricts the learning of high resolution features, which could potentially be useful in the later stages. In our network, we introduced an additional convolution with a kernel size of 3×3 before the first convolution layer in ResNet. This enables the network to learn more high resolution features without significantly increasing the inference time.

b) Higher Resolution Outputs (HR): Principally, downsampling reduces the resolution of feature maps and therefore generates coarser segmentations. Although deconvolution layers can upsample low resolution feature maps, it cannot recover all the details completely. Moreover, this procedure is not only computationally expensive but also memory intensive. To avoid downsampling, atrous, or also known as dilated-convolution [25], can be used. Atrous convolution "widens" the kernel and simulates a larger field of perception. For a 1-D input signal $x[i]$ with a filter $w[k]$ of length K, the atrous convolution is defined as:

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k]\, w[k] \qquad (1)$$

The rate r denotes the stride with which the input signal is sampled. A rate of 2 corresponds to a convolution on a 2×2 pooled feature map. In our proposed architecture, we change the last convolution with a stride of two to one in ResNet and every following convolution to an atrous convolution with $r = 2$. This way, the smallest resolution is not 32-times but 16-times downsampled and we keep the higher resolution details while aggregating the same amount of context as before.

c) Multiscale Blocks (MS): Every object in a scene can potentially differ in size and by distance. Filters learned by DCNNs are often not well suited for this multiscale appearance. This has lead to several investigations in the learning of multiscale features in DCNNs [7], [3]. These techniques often incorporate multi-resolution input images which again lead to higher computational cost. We propose a novel technique to efficiently generate multiscale features throughout the network without increasing the amount of parameters. A typical ResNet block consists of a 1×1 convolution followed by a 3×3 convolution and a 1×1 convolution as shown in Fig. 2. Our proposed multiscale Resnet block has in general a similar structure, but we change the 3×3 convolution as follows: we split the convolution into two separate convolutions with half the number of feature maps each. The first convolution has the same properties as before, while the second is an atrous convolution with $r > 1$. Both convolutions run in parallel and are then merged by concatenation which results in the same number of feature maps as without the splitting. This module enables the network to learn multiscale features in every block. Additionally, the concatenation preserves all features within the block so that the network can learn to combine features that have been generated on different scales. Our proposed multiscale blocks are shown in the legend in Fig. 2 (Bottom right two blocks).

We perform an in-depth analysis of the proposed improvements by initially taking the Resnet Upconv and evaluating the performance by incorporating the aforementioned techniques in every step. The results from this experiment are discussed in Sec IV-C.

B. Convoluted Mixture of Deep Experts

In order to fuse multiple spectra and modalities using expert networks, we propose the following framework as depicted in Fig. 3. Individual expert networks specializing in a particular subspace, map the representation of the input

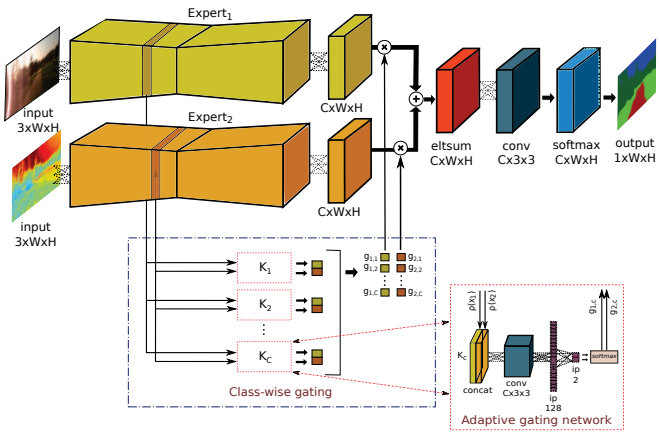

Fig. 3. Convoluted Mixture of Deep Experts framework. Any segmentation network can be plugged in for each of the experts depicted. We use the DCNN described in III-A for our experiments.

to a corresponding segmentation mask. The framework is adaptable to an arbitrary number of experts, for simplicity we consider two experts in our descriptions. The gating layer acts like a multiplexer, which maps outputs of N experts to a probabilistically fused representation. We train the gating network to learn a convex combination of experts by back-propagating into the weights, thus making them learnable parameters, similar to any other synapse weight or convolutional kernel. Complementary fused kernels are then further learned on top of this fusion to yield a robust pixel-wise segmentation output.

We represent the training set of a CMoDE with E experts and C segmentation classes, as $S = \{(X_n, y_n),\ n = 1,2,\ldots,N\}$. Each training example, $X_n = \{x_1, x_2, \ldots, x_E\}$ is a vector of raw images from different modalities, where image x_i is shown only to the i-th expert. y_n is a $W \times H$ segmentation mask, where, $y_n(r,c) \in \{0,1,\ldots,C\}$, for $r \in \{1,2,\ldots,W\}, c \in \{1,2,\ldots,H\}$, maps the membership of pixel $x_i(r,c)$ for each input representation, in one of the C classes.

Since experts train on images from different modalities or spectra, each one specializes in a particular sub-space of X_n. The i-th expert, produces its own segmentation mask, denoted by $h_i(x_i)$. The resultant output is a convex combination of the outputs of E experts; weighted by $g(X_n)$, which is the output of the adaptive gating network. Our network further learns fused kernels over these outputs using a convolution layer. The final output is a per-pixel segmentation mask $\hat{y}_n$, corresponding to the input X_n and is written as

$$\hat{y}_n = f(X_n) = softmax\left(\mathcal{W} * \left[\sum_{i=1}^{E} g_i(X_n) \cdot h_i(x_i)\right]\right) \quad (2)$$

where, $g_i(X_n)$ corresponds to the scalar weight for the i-th expert, $\mathcal{W}$ is a stack of convolution kernels learnt over the fused representation and $*$ is the convolution operation. Replacing the input to the gating network, X_n, consisting of raw pixels with a representation of X_n from within the expert network defined by $\rho(X_n) = (r(x_1), r(x_2), \ldots, r(x_E))$, yields improved results with lesser computation time. $r(x_i)$ is a representation of x_i taken from the the i-th expert, for instance, the output of *conv4*. Defining $\rho(\cdot)$ to be a representation from the contracting part of the expert network and leveraging the fact that H and W decrease, while channel depth increases towards the end of the contracting part, forcing the network to increase the "what" and reduce the "where" [19]. This "what" is of primary importance to the adaptive gating network. For instance, if an RGB image is washed out due to poor lighting conditions, the network needs to only know "what" and not "where" the image is washed out; making it rely less on the RGB expert and give it a lower score, $g_{RGB}(X_n)$, while relying more on other experts. Re-writing the Eq. 2 with $\rho(\cdot)$ as

$$\hat{y}_n = f(X_n) = softmax\left(\mathcal{W} * \left[\sum_{i=1}^{E} g_i(\rho(X_n)) \cdot h_i(x_i)\right]\right) \quad (3)$$

Each expert network is trained separately and uses a subspace of X_n containing images of a specific spectra or modality. The adaptive gating network takes $\rho(X_n)$ as input and produces probability distribution over the experts. The 3D volumes each of size $C \times W \times H$ from each expert are weighted according to $g_i(\rho(X_n))$. Convolutions followed by a *softmax* layer convert these to per-pixel class membership probabilities, allowing us to interpret $g_i(\rho(X_n))$ as the probability of choosing an expert. More formally

$$g_i(\rho(X_n)) = P(E_i \mid \rho(X_n)). \quad (4)$$

It should be noted that while training the adaptive gating network, the weights of the expert networks are kept constant. During testing the *softmax* is replaced with an *argmax* layer.

Now, we extend the concept of CMoDE further by reconstructing the adaptive gating network so that it generates a vector of weights for each expert, where each element in the vector corresponds to how much the gating trusts the class-specific kernels learned by that expert. Specifically, this vector represents a probability distribution over the C classes for the i-th expert. We can now represent the class-specific CMoDE as

$$\hat{y}_n = softmax\left(\mathcal{W} * \left[\sum_{i=1}^{E}\sum_{c=1}^{C} g_{i,c}(\rho(X_n)) \cdot h_{i,c}(x_i)\right]\right) \quad (5)$$

Here, instead of choosing a convex combination over the output of the experts, we choose a convex combination over the class-specific outputs of the experts. Therefore, generalizing Eq. 4, we can interpret $g_{i,c}$ as a joint probability, namely

$$g_{i,c}(\rho(X_n)) = P(E_i, K_c \mid \rho(X_n)). \quad (6)$$

This class-specific probability distribution gives the model the ability to choose different modalitites or spectra for different classes. However, due to the complexity of the adaptive gating network which is equipped with more degrees of freedom, training the network can become cumbersome and lead to overfitting in the gating network. We add spatial dropout [23] and rectified linear units (ReLUs) after the convolution layer in the gating network to avoid overfitting.

C. Network Training

We train the CMoDE network using a multi-stage training scheme. We first train the expert network to produce the respective segmentation masks in the datasets, followed by training the class-specific gating network and the fused

convolutions, by keeping the weights of the expert network constant. This forces the gating network to use the representations learned by the experts and leverages complementary features from the experts. We train the networks with a initial learning rate $\lambda_0 = 10^{-6}$ and with the poly learning rate policy as, $\lambda_n = \lambda_0 \times \left(\frac{1-N}{N_{\max}}\right)^c$, where λ_n is the current learning rate, N is the iteration number, $N_{\max}$ is the maximum number of iterations and c is the power. We train using Stochastic Gradient Decent (SGD) with a momentum of 0.9 and a mini-batch of 4 for $40,000$ iterations. The goal is to learn features by minimizing the cross-entropy $(softmax)$ loss that can be computed as $\mathscr{L}(u,y) = -\sum_k y_k \log u_k$.

IV. Experimental Results

We use the Caffe [11] deep learning library with cuDNN backend for our implementations. The metrics reported in this paper correspond to Mean Intersection-over-Union (IoU), Average Precision (AP), False Positive Rate (FPR) and False Negative Rate (FNR), as used in the PASCAL VOC challenges.

A. Datasets and Augmentation

We evaluate the AdapNet architecture along with our CMoDE framework on three publicly available datasets containing diverse environments ranging from forested landscapes to urban city streets: namely, Freiburg Multispectral dataset [24], Cityscapes [4] and Synthia [20]. The datasets were specifically chosen with the criteria that they should contain commonly observed environmental appearance changes and seasonal variations.

We use the adverse environments set from the Freiburg Multispectral forest dataset containing 6 classes: Sky, Obstacles, Road, Grass, Vegetation, Background and Void. This dataset contains multispectral and multimodal images of forested environments with varying conditions such as low-lighting, snow, glare and motion blur. We select RGB, depth and EVI (Enhanced Vegetation Index) as the modalities to experiment on. We also benchmark on two urban city datasets, Cityscapes and Synthia. The Cityscapes dataset contains RGB and depth images from over 50 cities with varying seasons, time of the day and weather conditions. On the other hand, the Synthia dataset contains photo-realistic images rendered from a virtual city with different view points, multiple seasons, weather and lighting conditions including rain, snow, dusk, sunset and night scenes. This dataset is extremely challenging due to the presence of several dynamic objects distant from the camera and objects such as posts, signs and fences that have a small footprint in the image. The Synthia dataset contains 13 classes, namely: Sky, Building, Road, Sidewalk, Fence, Vegetation, Pole, Car/Truck/Bus, Traffic Sign, Pedestrians, Rider/Bicycle/Motorbike and Background. Recent work [20] has shown improved performance in segmenting scenes by training models on the concatenation of synthetic and real datasets. In order to evaluate the adaptability of this transfer learning, we combine classes in the Cityscapes dataset to yield the same categories as in the Synthia dataset. In order to facilitate benchmarking, we provide the file paths for all these mappings of classes, as well as the train and validation splits on the project page: `http://deepscene.cs.uni-freiburg.de/`.

TABLE I

Performance comparison of AdapNet to baseline models.

Network	Dataset	IoU	AP	FPR	FNR
FCN8 [15]	Freiburg Forest	77.46	87.38	10.32	12.12
SegNet [1]	Freiburg Forest	74.81	84.63	13.53	11.65
ParseNet [14]	Freiburg Forest	83.65	90.07	8.94	7.41
UpNet [17]	Freiburg Forest	84.90	91.16	7.80	7.40
AdapNet (ours)	**Freiburg Forest**	**88.25**	**93.38**	**6.08**	**5.67**
FCN8 [15]	Cityscapes	64.57	77.62	15.62	19.80
SegNet [1]	Cityscapes	47.78	61.21	31.25	20.97
ParseNet [14]	Cityscapes	65.61	78.52	15.57	18.82
UpNet [17]	Cityscapes	62.62	75.53	16.50	20.87
AdapNet (ours)	**Cityscapes**	**69.39**	**81.72**	**13.57**	**17.04**
FCN8 [15]	Synthia-Cityscapes	65.24	78.48	13.73	21.03
SegNet [1]	Synthia-Cityscapes	27.10	42.26	46.96	25.94
ParseNet [14]	Synthia-Cityscapes	68.87	81.09	12.63	18.50
UpNet [17]	Synthia-Cityscapes	65.78	75.94	17.91	16.30
AdapNet (ours)	**Synthia-Cityscapes**	**72.91**	**84.31**	**10.85**	**16.23**

We perform a series of augmentations on the training images to provide the network with more training data and additional prior knowledge about variations in the scene. We randomly apply the following augmentations: rotation, translation, skewing, scaling, vignetting, cropping, flipping, color, brightness and contrast modulation.

B. Baseline Comparison

In this section, we present comprehensive evaluations of our AdapNet architecture in comparison to state-of-the-art models on three standard benchmark datasets as described in Sec. IV-A. Our focus is on comparison with end-to-end architectures without additional post-processing and that have fast run-times. Tab. I shows the results from these experiments performed using the RGB images from the benchmarks. Our AdapNet architecture with about 91 layers manages to overcome the vanishing gradient and optimization problems, while achieving state-of-the-art performance on all the three challenging benchmarks. The AdapNet model achieves an IoU of 88.25% on the Freiburg Multispectral Forest dataset, which constitutes to an improvement of 3.35% over UpNet (also known as FastNet), which was the previous state-of-the-art. On Cityscapes and Synthia benchmarks, our model achieves an improvement of 3.78% and 4.04% respectively, in comparison to the best performing baselines. This improvement can be attributed to the highly representational multiscale features learned by our model which enable the segmentation of very distant objects present in Synthia and Cityscapes. Moreover this also facilitates the reduction of false positives substantially as shown in Tab. I.

For robotic perception, it is critical that the network performs inference in near real-time to enable quick decision making. Although our model is deeper than the current state-of-the-art, the fewer parameters enables the architecture to perform inference in near real-time. Fig. 4 shows comparisons of forward-pass time's on the NVIDIA TITAN X.

C. Analysis of the AdapNet Architecture

We evaluated each configuration of our proposed AdapNet architecture on the Cityscapes dataset, as detailed in Sec. III-

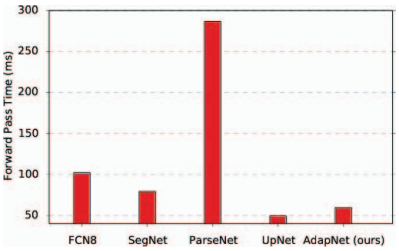

Fig. 4. Comparison of forward-pass timing for an input image of size 768×384. The baselines ParseNet and FCN8s take more than 100ms, which makes it difficult to use for robotic perception. In contrast, AdapNet takes about 59ms, while achieving state-of-the-art performance.

TABLE II

ANALYSIS OF ADAPNET ON THE CITYSCAPES DATASET.

Approach	IoU	FPR	FNR
ResNet Upconv	63.56	15.53	20.60
ResNet Upconv + FC	64.02	15.44	20.54
ResNet Upconv + FC + HR	65.15	14.83	20.11
ResNet Upconv + FC + HR + MS	69.39	13.57	17.04

A.1. We chose the Cityscapes dataset, due to the balanced and relevance of classes. Furthermore, the dataset includes varying conditions and thin structures such as poles and fences that appear in far and near distances with respect to the viewing perspective, thereby making it highly challenging and suitable for testing multiscale perception. Results from this experiment are shown in Tab. II. Our baseline configuration denoted as ResNet Upconv which constitutes the ResNet-50 architecture with deconvolutions, yields a mean IoU of 63.56. However, this performance is lower than than the best performing Parsenet baseline architecture. By adding the additional convolution (FC) in the beginning of the network, yields an improved IoU of 64.02. This demonstrates that performing more convolutions on the higher resolution feature maps, improves the over all segmentation accuracy. We further improve this model by removing the last pooling layer and replacing the convolution layers that follow with atrous convolutions. This model achieves a mean IoU of 65.15. We then add multiscale blocks (MS) as described in Sec. III-A.1 which remarkably boosts the mean IoU to 69.39. This model exceeds the performance of the state-of-the-art models that we evaluated by a substantial margin.

D. Fusion Experiments

In this section, we present a thorough comparison of our CMoDE fusion framework using the spectra and modalities contained in the three challenging benchmarks. For a baseline, we show the performance obtained by averaging both expert networks trained on a specific modality. We compare the performance of CMoDE with two other deep fusion techniques: late-fusion and LFC [24]. In the late-fusion approach we add a 1×1 convolution layer after last layer of the expert network and then sum the feature maps element-wise, followed by a *softmax* classifier. However, the late-fusion approach does not perform better than averaging the predictions as seen in Tab. III, IV and V. This is primarily due to the inability to learn kernels on top of the fusion,

TABLE III

COMPARISON OF FUSION ON THE FREIBURG FOREST DATASET.

Input	Approach	IoU	AP	FPR	FNR
RGB	Unimodal	88.25	93.38	6.08	5.67
DEPTH	Unimodal	79.96	88.37	10.05	9.99
EVI	Unimodal	85.51	92.25	6.93	7.54
RGB-D	Average	86.41	92.48	6.88	6.71
	Late fusion	86.59	92.28	7.13	6.28
	LFC [24]	89.31	94.08	5.24	5.44
	CMoDE	**90.12**	**94.45**	**5.12**	**5.39**
RGB-E	Average	88.23	93.61	5.88	5.89
	Late fusion	88.13	93.58	5.92	5.94
	LFC [24]	88.97	93.85	5.64	5.98
	CMoDE	**91.06**	**95.98**	**5.02**	**5.24**

TABLE IV

COMPARISON OF FUSION ON THE CITYSCAPES DATASET.

Input	Approach	IoU	AP	FPR	FNR
RGB	Unimodal	69.39	81.72	13.57	17.04
DEPTH	Unimodal	59.25	74.74	17.28	23.47
RGB-D	Average	68.86	84.20	11.48	19.66
	Late fusion	67.98	79.71	15.51	16.51
	LFC [24]	69.25	85.28	10.03	18.72
	CMoDE	**71.72**	**89.98**	**9.66**	**14.62**

thereby the models do not perform better than the individual experts themselves. The LFC on the other hand performs better than the averaging approach and unimodal segmentation due to its ability to learn complementary features over the fusion. However, our proposed CMoDE fusion scheme outperforms LFC in all the three benchmarks and combinations of modalities. This can be attributed to its ability to choose the relevant class-specific kernels before the fusion, which is unrealizable in the LFC scheme.

In order to further evaluate the performance, we show qualitative comparisons in segmentation using unimodal RGB and the CMoDE fusion scheme in Fig. 5. In the scenes from the Synthia datasets shown in Fig. 5(a) and 5(b), the improved performance of AdapNet with CMoDE can be especially seen in segmenting thin structures such as poles, fences, traffic lights and pedestrians. While in the scenes from Cityscapes (Fig. 5(c) and 5(d)), the accuracy in segmentation can be seen in classes such as roads made of cobblestones, pavements and pedestrians. Furthermore, in contrast to the unimodal model, the CMoDE reliably detects the edges of the sidewalks and vegetation. Finally, Fig. 5(e) and 5(f) show some examples from the Freiburg Multispectral Forest benchmark. It can be seen that the fused features from CMoDE helps segment the trail more accurately even in the presence of disturbances such as glare on the optics and snow. The model also consistently avoids outliers like misclassified vegetation. In Fig. 1, we further show qualitative comparison of segmentation from AdapNet with LFC and AdapNet with CMoDE. These results underline the robustness and the adaptivity of our proposed network and fusion scheme.

TABLE V

COMPARISON OF FUSION ON THE SYNTHIA-CITYSCAPES DATASET.

Input	Approach	IoU	AP	FPR	FNR
RGB	Unimodal	72.91	84.31	10.85	16.23
DEPTH	Unimodal	70.74	83.82	10.61	18.64
RGB-D	Average	74.11	87.97	8.16	17.22
	Late fusion	68.73	88.37	6.27	24.99
	LFC [24]	75.12	89.95	7.95	13.94
	CMoDE	**77.11**	**90.28**	**7.02**	**12.87**

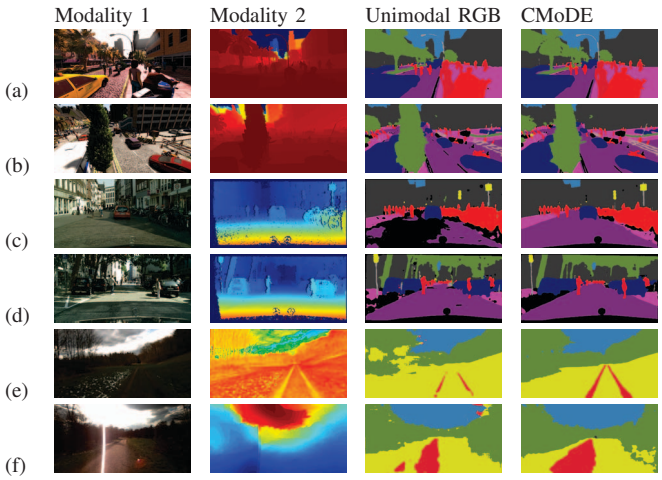

Fig. 5. Comparison between segmentation using only RGB versus CMoDE fusion on the datasets: Synthia, Cityscapes and Freiburg Forest. a) and b) show street scenes from the synthia dataset. Note that the CMoDE fusion performs better for fine structures like fences and poles. c) and d) show street scenes from the cityscapes dataset. CMoDE detects streets out of cobblestone and sidewalks more accurate. e) and f) show segmented images from the freiburg forest dataset. One can note that the multimodal fusion leads to better segmentation of the path and avoids misclassification of vegetation.

E. Robustness Evaluation

In order to further analyse the performance in specific adverse conditions we use the Synthia-Sequences dataset that contains several sequences in varying seasons, environmental conditions and scenarios. Specifically, we train on RGB and depth images using our AdapNet model on the combination of sequences 5, 6 and 7, which correspond to a city-like environment, highway and a European town. We present both qualitative and quantitative results of testing this model on Synthia-Sequences 1,2, and 4, in conditions such as fall, winter, summer, dawn, night, rain and sunset. Fig. 6 shows the results from this comparison and the qualitative results are shown in Fig. 7. In addition, we also show segmentation in the presence of snow and shadows from the Freiburg Forest dataset and on images from Cityscapes which involve motion blur.

In addition, we performed real-world experiments of autonomously traversing 4.52 km in a challenging forested environment using our VIONA robot as shown in Fig. 8. We implemented our segmentation pipeline using ROS and the Caffe library on the NVIDIA TX1 embedded GPU which has 256 CUDA cores. We first capture the scene using the Bumblebee2 stereo cameras on the robot and segment the images with our AdapNet model trained on the Multispectral

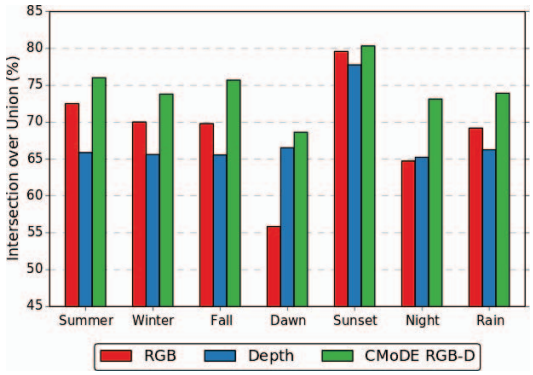

Fig. 6. Performance of the CMoDE fusion scheme on the Synthia Sequences dataset in various seasons and conditions.

Freiburg dataset. We then compute waypoints for the robot to follow from the segmentation masks in the camera frame and subsequently forward them to the planner, which then executes the trajectory. The robot navigated with an average speed of $0.9\,\mathrm{m\,s^{-1}}$, while a forward pass on the NVIDIA TX1 embedded GPU took about 623 ms. During the entire experiment the robot did not have any prior map of the area and just used the segmentation on the fly to traverse the trail. The robot encountered several challenging situations during the experiment including low-lighting due to forest canopy, occasional glare from the sun, shadows from trees and motion blur. The perception system consisting of the AdapNet model was robust to all these disturbances and performed inference online that enabled the successful autonomous run. Videos from this experiment and a live demo of various models presented in this paper can be accessed at `http://deepscene.cs.uni-freiburg.de/`.

V. CONCLUSION

In this paper, we introduced a novel end-to-end semantic segmentation architecture complemented with an adaptive fusion strategy for robust semantic segmentation. Our proposed fusion scheme is independent of the base expert architecture and can applied to arbitrary number of experts that specialize on a subset of the input space. Our AdapNet architecture outperforms other end-to-end semantic segmentation networks and our class-specific fusion scheme achieves state-of-the-art performance compared to unimodal segmentation and existing fusion techniques. We demonstrated the robustness of our approach on three different publicly available datasets in diverse environments and conditions. Additionally, we presented experimental results from autonomously navigating 4.52 km in a forested environment using only the segmentation for perception.

REFERENCES

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv: 1511.00561*, 2015.

[2] M. Bojarski *et al.*, "End to end learning for self-driving cars," *arxiv preprint arXiv: 1604.07316*, 2016.

[3] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arxiv preprint arXiv: 1606.00915*, 2016.

[4] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

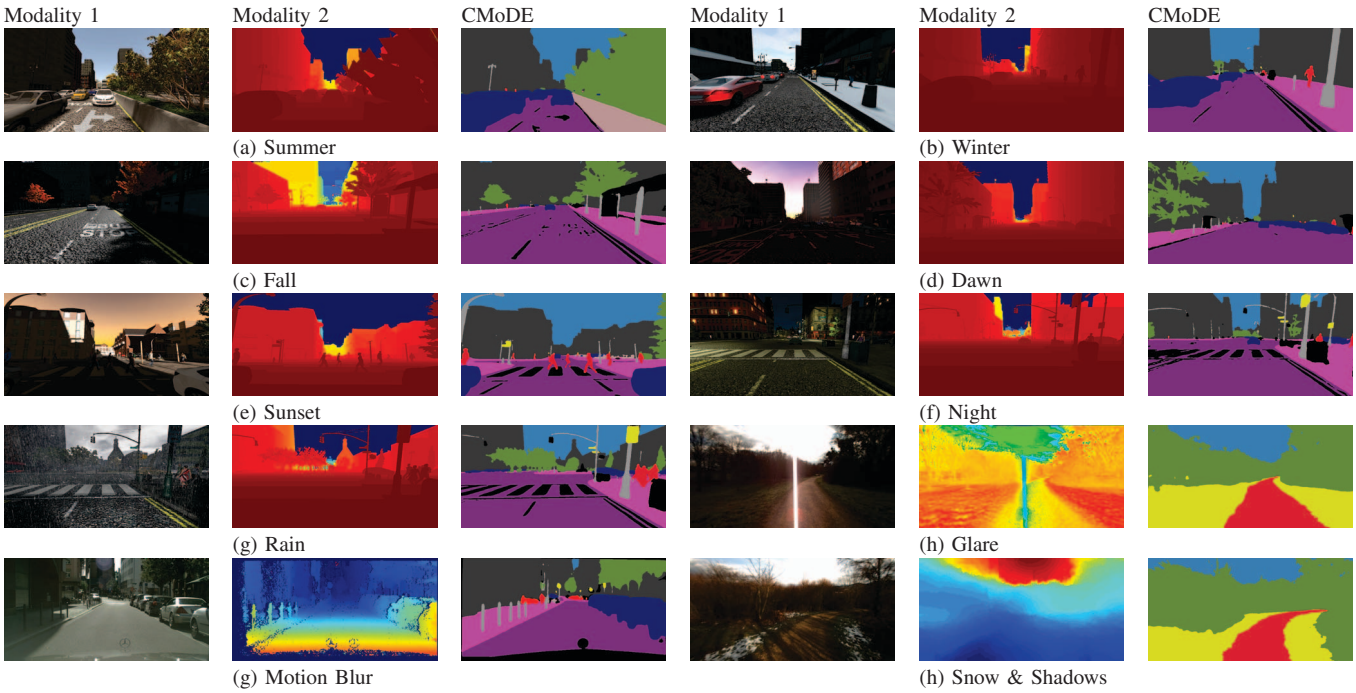

Fig. 7. Qualitative evaluation of segmentation in adverse conditions. Our CMoDE fusion model adapts to the changing conditions, thereby yielding robust segmentation even in the presence on disturbances such as low lighting, rain, snow, motion blur and glare.

(a) The robot used in our experiments, equipped with four stereo cameras.

(b) Trajectory taken by the robot during the autonomous run in the forest.

(c) Planning interface showing the waypoints, camera images and the corresponding segmentation.

Fig. 8. Our robot autonomously navigating in a forested environment for 4.52 km, using only stereo cameras for perception. The segmentation is performed on the NVIDIA TX1 embedded platform and the waypoints derived from the segmentation are sent to the autonomy system on the robot.

[5] D. Eigen, M. Ranzato, and I. Sutskever, "Learning factored representations in a deep mixture of experts," *ICLR workshop*, 2014.

[6] A. Eitel *et al.*, "Multimodal deep learning for robust rgb-d object recognition," in *IROS*, 2015.

[7] G. Ghiasi and C. C. Fowlkes, "Laplacian reconstruction and refinement for semantic segmentation," in *ECCV*, 2016.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2015.

[9] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *CVPR*, 2015.

[10] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural computation*, vol. 3, no. 1, pp. 79–87, 1991.

[11] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[12] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," *arxiv preprint arXiv: 1606.00373*, 2016.

[13] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *IJRR*, vol. 34, no. 4-5, pp. 705–724, 2015.

[14] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv: 1506.04579*, 2015.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015.

[16] O. Mees, A. Eitel, and W. Burgard, "Choosing smartly: Adaptive multimodal fusion for object detection in changing environments," in *IROS*, 2016.

[17] G. Oliveira, W. Burgard, and T. Brox, "Efficient deep methods for monocular road segmentation," in *IROS*, 2016.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[19] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[20] G. Ros *et al.*, "The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *CVPR*, 2016.

[21] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv: 1409.1556*, 2014.

[23] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *CVPR*, 2015, pp. 648–656.

[24] A. Valada, G. Oliveira, T. Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in *ISER*, Oct. 2016.

[25] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *ICLR*, 2016.