



# Self-Supervised Model Adaptation for Multimodal Semantic Segmentation

Abhinav Valada<sup>1</sup> · Rohit Mohan<sup>1</sup> · Wolfram Burgard<sup>1,2</sup>

Received: 24 July 2018 / Accepted: 5 June 2019 / Published online: 8 July 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Learning to reliably perceive and understand the scene is an integral enabler for robots to operate in the real-world. This problem is inherently challenging due to the multitude of object types as well as appearance changes caused by varying illumination and weather conditions. Leveraging complementary modalities can enable learning of semantically richer representations that are resilient to such perturbations. Despite the tremendous progress in recent years, most multimodal convolutional neural network approaches directly concatenate feature maps from individual modality streams rendering the model incapable of focusing only on the relevant complementary information for fusion. To address this limitation, we propose a multimodal semantic segmentation framework that dynamically adapts the fusion of modality-specific features while being sensitive to the object category, spatial location and scene context in a self-supervised manner. Specifically, we propose an architecture consisting of two modality-specific encoder streams that fuse intermediate encoder representations into a single decoder using our proposed self-supervised model adaptation fusion mechanism which optimally combines complementary features. As intermediate representations are not aligned across modalities, we introduce an attention scheme for better correlation. In addition, we propose a computationally efficient unimodal segmentation architecture termed AdapNet++ that incorporates a new encoder with multiscale residual units and an efficient atrous spatial pyramid pooling that has a larger effective receptive field with more than  $10 \times$  fewer parameters, complemented with a strong decoder with a multi-resolution supervision scheme that recovers high-resolution details. Comprehensive empirical evaluations on Cityscapes, Synthia, SUN RGB-D, ScanNet and Freiburg Forest benchmarks demonstrate that both our unimodal and multimodal architectures achieve state-of-the-art performance while simultaneously being efficient in terms of parameters and inference time as well as demonstrating substantial robustness in adverse perceptual conditions.

**Keywords** Semantic segmentation · Multimodal fusion · Scene understanding · Model adaptation · Deep learning

## 1 Introduction

Humans have the remarkable ability to instantaneously recognize and understand a complex visual scene which has

piqued the interest of computer vision researchers to model this ability since the 1960s (Fei-Fei et al. 2004). There are numerous ever-expanding applications to this capability ranging from robotics (Xiang and Fox 2017; Boniardi et al. 2019; Radwan et al. 2018b) and remote sensing (Audebert et al. 2018) to medical diagnostics (Ronneberger et al. 2015) and content-based image retrieval (Noh et al. 2017). However, there are several challenges imposed by the multifaceted nature of this problem including the large variation in types and scales of objects, clutter and occlusions in the scene as well as outdoor appearance changes that take place throughout the day and across seasons.

Deep Convolutional Neural Network (DCNN) based methods (Long et al. 2015; Chen et al. 2016; Yu and Koltun 2016) modelled as a Fully Convolutional Neural Network (FCN) have dramatically increased the performance on sev-

---

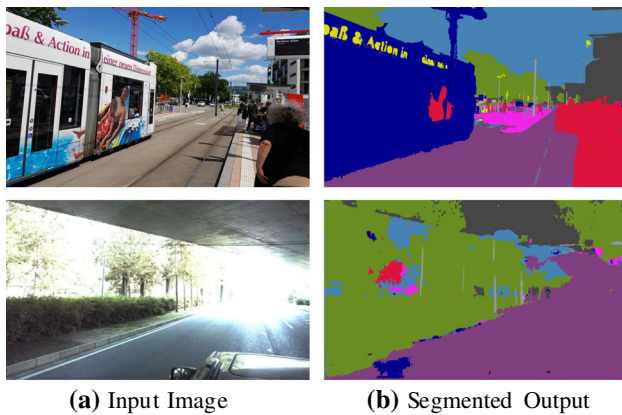
Communicated by Anelia Angelova, Gustavo Carneiro, Niko Sünderhauf, Jürgen Leitner.

---

✉ Abhinav Valada  
valada@cs.uni-freiburg.de  
Rohit Mohan  
mohan@cs.uni-freiburg.de  
Wolfram Burgard  
burgard@cs.uni-freiburg.de

<sup>1</sup> University of Freiburg, Freiburg, Germany

<sup>2</sup> Toyota Research Institute, Los Altos, USA



**Fig. 1** Example real-world scenarios where current state-of-the-art approaches demonstrate misclassifications. The first row shows an issue of mismatched relationship as well as inconspicuous classes where a decal on the train is falsely predicted as a person and the decal text is falsely predicted as a sign. The second row shows misclassifications caused by overexposure of the camera due to car exiting a tunnel (Color figure online)

eral semantic segmentation benchmarks. Nevertheless, they still face challenges due to the diversity of scenes in the real-world that cause mismatched relationship and inconspicuous object classes. Figure 1 shows two example scenes from real-world scenarios in which misclassifications are produced due to the decal on the train which is falsely predicted as a person and a traffic sign (first row), and overexposure of the camera caused by the vehicle exiting a tunnel (second row). In order to accurately predict the elements of the scene in such situations, features from complementary modalities such as depth and infrared can be leveraged to exploit properties such as geometry and reflectance, respectively. Moreover, the network can exploit complex intra-modal dependencies more effectively by directly learning to fuse visual appearance information from RGB images with learned features from complementary modalities in an end-to-end fashion. This not only enables the network to resolve inherent ambiguities and improve reliability but also obtain a more holistic scene segmentation.

While most existing work focuses on where to fuse modality-specific streams topologically (Hazirbas et al. 2016; Schneider et al. 2017; Valada et al. 2016c) and what transformations can be applied on the depth modality to enable better fusion with visual RGB features (Gupta et al. 2014; Eitel et al. 2015), it still remains an open question as to how to enable the network to dynamically adapt its fusion strategy based on the nature of the scene such as the types of objects, their spatial location in the world and the present scene context. This is a crucial requirement in applications such as robotics and autonomous driving where these systems run in continually changing environmental contexts. For example, an autonomous car navigating in ideal weather conditions can primarily rely on visual information but when

it enters a dark tunnel or exits an underpassage, the cameras might experience under/over exposure, whereas the depth modality will be more informative. Furthermore, the strategy to be employed for fusion also varies with the types of objects in the scene, for instance, infrared might be more useful to detect categories such as people, vehicles, vegetation and boundaries of structures but it does not provide much information on object categories such as the sky. Additionally, the spatial location of objects in the scene also has an influence, for example, the depth modality provides rich information on objects that are at nearby distances but degrades very quickly for objects that are several meters away. More importantly, the approach employed should be robust to sensor failure and noise as constraining the network to always depend on both modalities and use noisy information can worsen the actual performance and lead to disastrous situations.

Due to these complex interdependencies, naively treating modalities as multi-channel input data or concatenating independently learned modality-specific features does not allow the network to adapt to the aforementioned situations dynamically. Moreover, due to the nature of this dynamicity, the fusion mechanism has to be trained in a self-supervised manner in order to make the adaptivity emergent and to generalize effectively to different real-world scenarios. As a solution to this problem, we present the Self-Supervised Model Adaptation (SSMA) fusion mechanism that adaptively recalibrates and fuses modality-specific feature maps based on the object class, its spatial location and the scene context. The SSMA module takes intermediate encoder representations of modality-specific streams as input and fuses them probabilistically based on the activations of individual modality streams. As we model the SSMA block in a fully convolutional fashion, it yields a probability for each activation in the feature maps which represents the optimal combination to exploit complementary properties. These probabilities are then used to amplify or suppress the representations of the individual modality streams, followed by the fusion. As we base the fusion on modality-specific activations, the fusion is intrinsically tolerant to sensor failure and noise such as missing depth values.

Our proposed architecture for multimodal segmentation consists of individual modality-specific encoder streams which are fused both at mid-level stages and at the end of the encoder streams using our SSMA blocks. The fused representations are input to the decoder at different stages for upsampling and refining the predictions. Note that only the multimodal SSMA fusion mechanism is self-supervised, the semantic segmentation is trained in a supervised manner. We employ a combination of mid-level and late-fusion as several experiments have demonstrated that fusing semantically meaningful representations yields better performance in comparison to early fusion (Eitel et al. 2015; Valada et al. 2016b; Hazirbas et al. 2016; Xiang and Fox 2017; Radwan

et al. 2018a). Moreover, studies of the neural dynamics of the human brain has also shown evidence of late-fusion of modalities for recognition tasks (Cichy et al. 2016). However, intermediate network representations are not aligned across modality-specific streams. Hence, integrating fused multimodal mid-level features into high-level features requires explicit prior alignment. Therefore, we propose an attention mechanism that weighs the fused multimodal mid-level skip features with spatially aggregated statistics of the high-level decoder features for better correlation, followed by channel-wise concatenation.

As our fusion framework necessitates individual modality-specific encoders, the architecture that we employ for the encoder and decoder should be efficient in terms of the number of parameters and computational operations, as well as be able to learn highly discriminative deep features. State-of-the-art semantic segmentation architectures such as DeepLab v3 (Chen et al. 2017) and PSPnet (Zhao et al. 2017) employ the ResNet-101 (He et al. 2015a) architecture which consumes 42.39M parameters and 113.96B FLOPS, as the encoder backbone. Training such architectures requires a large amount of memory and synchronized training across multiple GPUs. Moreover, they have slow run-times rendering them impractical for resource constrained applications such as robotics and augmented reality. More importantly, it is infeasible to employ them in multimodal frameworks that require multiple modality-specific streams as we do in this work.

With the goal of achieving the right trade-off between performance and computational complexity, we propose the AdapNet++ architecture for unimodal segmentation. We build the encoder of our model based on the full pre-activation ResNet-50 (He et al. 2016) architecture and incorporate our previously proposed multiscale residual units (Valada et al. 2017) to aggregate multiscale features throughout the network without increasing the number of parameters. The proposed units are more effective in learning multiscale features than the commonly employed multigrid approach introduced in DeepLab v3 (Chen et al. 2017). In addition, we propose an efficient variant of the Atrous Spatial Pyramid Pooling (ASPP) (Chen et al. 2017) called eASPP that employs cascaded and parallel atrous convolutions to capture long range context with a larger effective receptive field, while simultaneously reducing the number of parameters by 87% in comparison to the originally proposed ASPP. We also propose a new decoder that integrates mid-level features from the encoder using multiple skip refinement stages for high resolution segmentation along the object boundaries. In order to aid the optimization and to accelerate training, we propose a multiresolution supervision strategy that introduces weighted auxiliary losses after each upsampling stage in the decoder. This enables faster convergence, in addition to improving the performance of the model along the object boundaries. Our

proposed architecture is compact and trainable with a large mini-batch size on a single consumer grade GPU.

Motivated by the recent success of compressing DCNNs by pruning unimportant neurons (Molchanov et al. 2017; Liu et al. 2017; Anwar et al. 2017), we explore pruning entire convolutional feature maps of our model to further reduce the number of parameters. Network pruning approaches utilize a cost function to first rank the importance of neurons, followed by removing the least important neurons and fine-tuning the network to recover any loss in accuracy. Thus far, these approaches have only been employed for pruning convolutional layers that do not have an identity or a projection shortcut connection. Pruning residual feature maps (third convolutional layer of a residual unit) also necessitates pruning the projected feature maps in the same configuration in order to maintain the shortcut connection. This leads to a significant drop in accuracy, therefore current approaches omit pruning convolutional filters with shortcut connections. As a solution to this problem, we propose a network-wide holistic pruning approach that employs a simple and yet effective strategy for pruning convolutional filters invariant to the presence of shortcut connections. This enables our network to further reduce the number of parameters and computing operations, making our model efficiently deployable even in resource constrained applications.

Finally, we present extensive experimental evaluations of our proposed unimodal and multimodal architectures on benchmark scene understanding datasets including Cityscapes (Cordts et al. 2016), Synthia (Ros et al. 2016), SUN RGB-D (Song et al. 2015), ScanNet (Dai et al. 2017) and Freiburg Forest (Valada et al. 2016b). The results demonstrate that our model sets the new state-of-the-art on all these benchmarks considering the computational efficiency and the fast inference time of 72 ms on a consumer grade GPU. More importantly, our dynamically adapting multimodal architecture demonstrates exceptional robustness in adverse perceptual conditions such as fog, snow, rain and night-time, thus enabling it to be employed in critical resource constrained applications such as robotics where not only accuracy but robustness, computational efficiency and run-time are equally important. To the best of our knowledge, this is the first multimodal segmentation work to benchmark on these wide range of datasets containing several modalities and diverse environments ranging from urban city driving scenes to indoor environments and unstructured forested scenes.

In summary, the following are the main contributions of this work:

1. A multimodal fusion framework incorporating our proposed SSMA fusion blocks that adapts the fusion of modality-specific features dynamically according to the object category, its spatial location as well as the scene context and learns in a self-supervised manner.

2. The novel AdapNet++ semantic segmentation architecture that incorporates our multiscale residual units, a new efficient ASPP, a new decoder with skip refinement stages and a multiresolution supervision strategy.
3. The eASPP for efficiently aggregating multiscale features and capturing long range context, while having a larger effective receptive field and over  $10 \times$  reduction in parameters compared to the standard ASPP.
4. An attention mechanism for effectively correlating fused multimodal mid-level and high-level features for better object boundary refinement.
5. A holistic network-wide pruning approach that enables pruning of convolutional filters invariant to the presence of identity or projection shortcuts.
6. Extensive benchmarking of existing approaches with the same input image size and evaluation setting along with quantitative and qualitative evaluations of our unimodal and multimodal architectures on five different benchmark datasets consisting of multiple modalities.
7. Implementations of our proposed architectures are made publicly available at <https://github.com/DeepSceneSeg> and a live demo on all the five datasets can be viewed at <http://deepscene.cs.uni-freiburg.de>.

## 2 Related Works

In the last decade, there has been a sharp transition in semantic segmentation approaches from employing hand engineered features with flat classifiers such as Support Vector Machines (Fulkerson et al. 2009), Boosting (Sturgess et al. 2009) or Random Forests (Shotton et al. 2008; Brostow et al. 2008), to end-to-end DCNN-based approaches (Long et al. 2015; Badrinarayanan et al. 2015). We first briefly review some of the classical methods before delving into the state-of-the-art techniques.

**Semantic Segmentation** Semantic segmentation is one of the fundamental problems in computer vision. Some of the earlier approaches for semantic segmentation use small patches to classify the center pixel using flat classifiers (Shotton et al. 2008; Sturgess et al. 2009) followed by smoothing the predictions using Conditional Random Fields (CRFs) (Sturgess et al. 2009). Rather than only relying on appearance based features, structure from motion features have also been used with randomized decision forests (Brostow et al. 2008; Sturgess et al. 2009). View independent 3D features from dense depth maps have been shown to outperform appearance based features, that also enabled classification of all the pixels in an image, as opposed to only the center pixel of a patch (Zhang et al. 2010). Plath et al. (2009) propose an approach to combine local and global features using a CRF and an image classification method. However,

the performance of these approaches is largely bounded by the expressiveness of handcrafted features which is highly scenario-specific.

The remarkable performance achieved by CNNs in classification tasks led to their application for dense prediction problems such as semantic segmentation, depth estimation and optical flow prediction. Initial approaches that employed neural networks for semantic segmentation still relied on patch-wise training (Grangier et al. 2009; Farabet et al. 2012; Pinheiro and Collobert 2014). Pinheiro and Collobert (2014) use a recurrent CNN to aggregate several low resolution predictions for scene labeling. Farabet et al. (2012) transforms the input image through a Laplacian pyramid followed by feeding each scale to a CNN for hierarchical feature extraction and classification. Although these approaches demonstrated improved performance over handcrafted features, they often yield a grid-like output that does not capture the true object boundaries. One of the first end-to-end approaches that learns to directly map the low resolution representations from a classification network to a dense prediction output was the Fully Convolutional Network (FCN) model (Long et al. 2015). FCN proposed an encoder-decoder architecture in which the encoder is built upon the VGG-16 (Simonyan and Zisserman 2014) architecture with the inner-product layers replaced with convolutional layers. While, the decoder consists of successive deconvolution and convolution layers that upsample and refine the low resolution feature maps by combining them with the encoder feature maps. The last decoder then yields a segmented output with the same resolution as the input image.

DeconvNet (Noh et al. 2015) propose an improved architecture containing stacked deconvolution and unpooling layers that perform non-linear upsampling and outperforms FCNs but at the cost of a more complex training procedure. The SegNet (Badrinarayanan et al. 2015) architecture eliminates the need for learning to upsample by reusing pooling indices from the encoder layers to perform upsampling. Oliveira et al. (2016) propose an architecture that builds upon FCNs and introduces more refinement stages and incorporates spatial dropout to prevent over fitting. The ParseNet (Liu et al. 2015) architecture models global context directly instead of only relying on the largest receptive field of the network. Recently, there has been more focus on learning multiscale features, which was initially achieved by providing the network with multiple rescaled versions of the image (Farabet et al. 2012) or by fusing features from multiple parallel branches that take different image resolutions (Long et al. 2015). However, these networks still use pooling layers to increase the receptive field, thereby decreasing the spatial resolution, which is not ideal for a segmentation network.

In order to alleviate this problem, Yu and Koltun (2016) propose dilated convolutions that allows for exponential



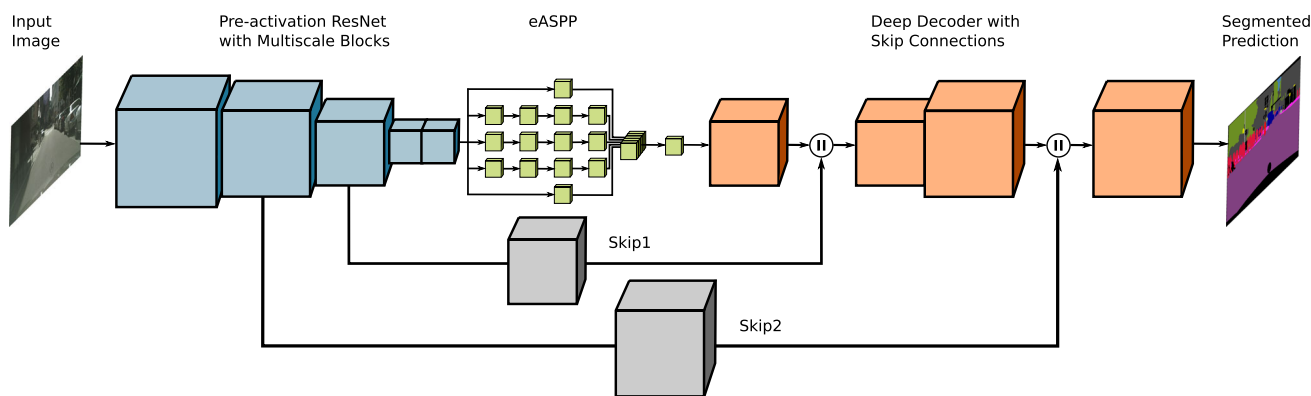
increase in the receptive field without decrease in resolution or increase in parameters. DeepLab (Chen et al. 2016) and PSPNet (Zhao et al. 2017) build upon the aforementioned idea and propose pyramid pooling modules that utilize dilated convolutions of different rates to aggregate multiscale global context. DeepLab in addition uses fully connected CRFs in a post processing step for structured prediction. However, a drawback in employing these approaches is the computational complexity and substantially large inference time even using modern GPUs that hinder them from being deployed in robots that often have limited resources. In our previous work (Valada et al. 2017), we proposed an architecture that introduces dilated convolutions parallel to the conventional convolution layers and multiscale residual blocks that incorporates them, which enables the model to achieve competitive performance at interactive frame rates. Our proposed multiscale residual blocks are more effective at learning multiscale features compared to the widely employed multigrid approach from DeepLab v3 (Chen et al. 2017). While in this work, we propose several new improvements for learning multiscale features, capturing long range context and improving the upsampling in the decoder, while simultaneously reducing the number of parameters and maintaining a fast inference time.

**Multimodal Fusion** The availability of low-cost sensors has encouraged novel approaches to exploit features from alternate modalities in an effort to improve robustness as well as the granularity of segmentation. Silberman et al. (2012) propose an approach based on SIFT features and MRFs for indoor scene segmentation using RGB-D images. Subsequently, Ren et al. (2012) propose improvements to the feature set by using kernel descriptors and by combining MRF with segmentation trees. Munoz et al. (2012) employ modality-specific classifier cascades that hierarchically propagate information and do not require one-to-one correspondence between data across modalities. In addition to incorporating features based on depth images, Hermans et al. (2014) propose an approach that performs joint 3D mapping and semantic segmentation using Randomized Decision Forests. There has also been work on extracting combined RGB and depth features using CNNs (Coupric et al. 2013; Gupta et al. 2014) for object detection and semantic segmentation. In most of these approaches, hand engineered or learned features are extracted from individual modalities and combined together in a joint feature set which is then used for classification.

More recently, there has been a series of DCNN-based fusion techniques (Eitel et al. 2015; Kim et al. 2017; Li et al. 2016) that have been proposed for end-to-end learning of fused representations from multiple modalities. These

fusion approaches can be categorized into early, hierarchical and late fusion methods. An intuitive early fusion technique is to stack data from multiple modalities channel-wise and feed it to the network as a four or six channel input. However, experiments have shown that this often does not enable the network to learn complementary features and cross-modal interdependencies (Valada et al. 2016b; Hazirbas et al. 2016). Hierarchical fusion approaches combine feature maps from multiple modality-specific encoders at various levels (often at each downsampling stage) and upsample the fused features using a single decoder (Hazirbas et al. 2016; Kim et al. 2017). Alternatively, Schneider et al. (2017) propose a mid-level fusion approach in which NiN layers (Lin et al. 2013) with depth as input are used to fuse feature maps into the RGB encoder in the middle of the network. Li et al. (2016) propose a Long-Short Term Memory (LSTM) context fusion model that captures and fuses contextual information from multiple modalities accounting for the complex interdependencies between them. (Qi et al. 2017) propose an interesting approach that employs 3D graph neural networks for RGB-D semantic segmentation that accounts for both 2D appearance and 3D geometric relations, while capturing long range dependencies within images.

In the late fusion approach, identical network streams are first trained individually on a specific modality and the feature maps are fused towards the end of network using concatenation (Eitel et al. 2015) or element-wise summation (Valada et al. 2016b), followed by learning deeper fused representations. However, this does not enable the network to adapt the fusion to changing scene context. In our previous work (Valada et al. 2016a), we proposed a mixture-of-experts CMoDE fusion scheme for combining feature maps from late fusion based architectures. Subsequently, in (Valada et al. 2017) we extended the CMoDE framework for probabilistic fusion accounting for the types of object categories in the dataset which enables more flexibility in learning the optimal combination. Nevertheless, there are several real-world scenarios in which class-wise fusion is not sufficient, especially in outdoor scenes where different modalities perform well in different conditions. Moreover, the CMoDE module employs multiple softmax loss layers for each class to compute the probabilities for fusion which does not scale for datasets such as SUN RGB-D which has 37 object categories. Motivated by this observation, in this work, we propose a multimodal semantic segmentation architecture incorporating our SSMA fusion module that dynamically adapts the fusion of intermediate network representations from multiple modality-specific streams according to the object class, its spatial location and the scene context while learning the fusion in a self-supervised fashion.



**Fig. 2** Overview of our proposed AdapNet++ architecture. Given an input image, we use the full pre-activation ResNet-50 architecture augmented with our proposed multiscale residual blocks to yield a feature map 16-times downsampled with respect to the input image resolution, then our proposed efficient atrous spatial pyramid (eASPP) module is

employed to further learn multiscale features and to capture long range context. Finally, the output of the eASPP is fed into our proposed deep decoder with skip connections for upsampling and refining the semantic pixel-level prediction (Color figure online)

### 3 AdapNet++ Architecture

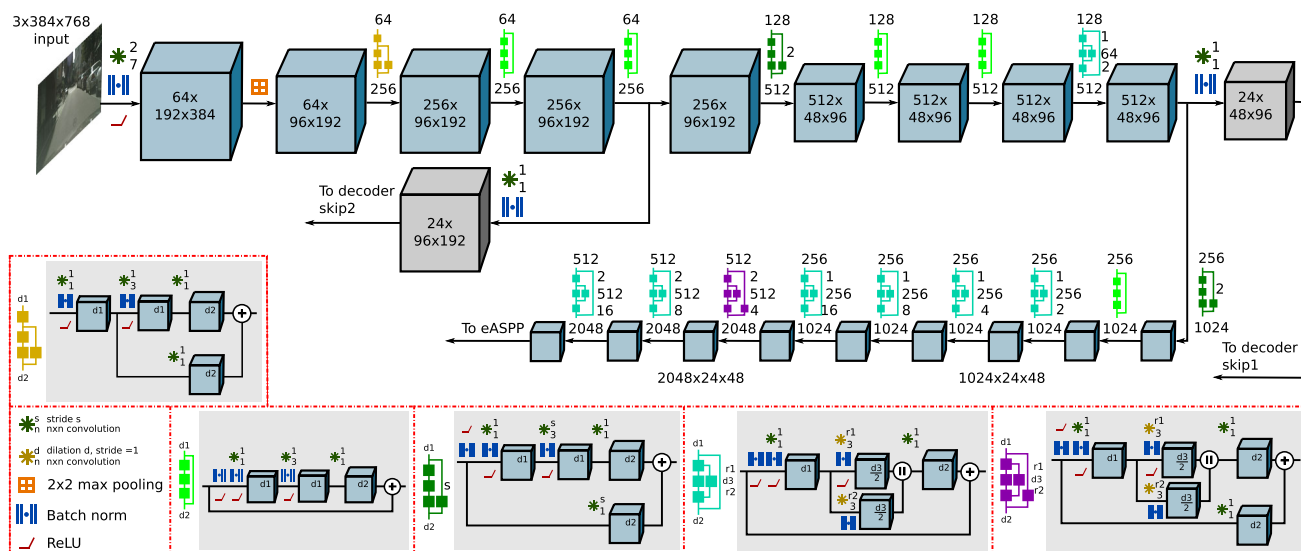
In this section, we first briefly describe the overall topology of the proposed AdapNet++ architecture and our main contributions motivated by our design criteria. We then detail each of the constituting architectural components and our model compression technique.

Our network follows the general fully convolutional encoder-decoder design principle as shown in Fig. 2. The encoder (depicted in blue) is based on the full pre-activation ResNet-50 (He et al. 2016) model as it offers a good trade-off between learning highly discriminative deep features and the computational complexity required. In order to effectively compute high resolution feature responses at different spatial densities, we incorporate our recently proposed multiscale residual units (Valada et al. 2017) at varying dilation rates in the last two blocks of the encoder. In addition, to enable our model to capture long-range context and to further learn multiscale representations, we propose an efficient variant of the atrous spatial pyramid pooling module known as eASPP which has a larger effective receptive field and reduces the number of parameters required by over 87% compared to the originally proposed ASPP in DeepLab v3 (Chen et al. 2017). We append the proposed eASPP after the last residual block of the encoder, shown as green blocks in Fig. 2. In order to recover the segmentation details from the low spatial resolution output of the encoder section, we propose a new deep decoder consisting of multiple deconvolution and convolution layers. Additionally, we employ skip refinement stages that fuse mid-level features from the encoder with the upsampled decoder feature maps for object boundary refinement. Furthermore, we add two auxiliary supervision branches after each upsampling stage to accelerate training and improve the gradient propagation in the network. We depict the decoder as orange blocks and the skip refinement

stages as gray blocks in the network architecture shown in Fig. 2. In the following sections, we discuss each of the aforementioned network components in detail and elaborate on the design choices.

#### 3.1 Encoder

Encoders are the foundation of fully convolutional neural network architectures. Therefore, it is essential to build upon a good baseline that has a high representational ability conforming with the computational budget. Our critical requirement is to achieve the right trade-off between the accuracy of segmentation and inference time on a consumer grade GPU, while keeping the number of parameters low. As we also employ the proposed architecture for multimodal fusion, our objective is to design a topology that has a reasonable model size so that two individual modality-specific networks can be trained in a fusion framework and deployed on a single GPU. Therefore, we build upon the ResNet-50 architecture with the full preactivation residual units (He et al. 2016) instead of the originally proposed residual units (He et al. 2015a) as they have been shown to reduce overfitting, improve the convergence and also yield better performance. The ResNet-50 architecture has four computational blocks with varying number of residual units. We use the bottleneck residual units in our encoder as they are computationally more efficient than the baseline residual units and they enable us to build more complex models that are easily trainable. The output of the last block of the ResNet-50 architecture is 32-times downsampled with respect to the input image resolution. In order to increase the spatial density of the feature responses and to prevent signal decimation, we set the stride of the convolution layer in the last block (*res4a*) from two to one which makes the resolution of the output feature maps 1/16-times the input image resolution. We then replace the



**Fig. 3** The proposed encoder is built upon the full pre-activation ResNet-50 architecture. Specifically, we remove the last downsampling stage in ResNet-50 by setting the stride from two to one, therefore the final output of the encoder is 16-times downsampled with respect to the input. We then replace the residual units that follow the last down-

sampling stage with our proposed multiscale residual units. The legend enclosed in red lines show the original pre-activation residual units in the bottom left (yellow, light green and dark green), while our proposed multiscale residual units are shown in the bottom right (Cyan and purple) (Color figure online)

residual blocks that follow this last downsampling stage with our proposed multiscale residual units that incorporate parallel atrous convolutions (Yu and Koltun 2016) at varying dilation rates.

A naive approach to compute the feature responses at the full image resolution would be to remove the downsampling and replace all the convolutions to atrous convolutions having a dilation rate  $r \geq 2$  but this would be both computation and memory intensive. Therefore, we propose a novel multiscale residual unit (Valada et al. 2017) to efficiently enlarge the receptive field and aggregate multiscale features without increasing the number of parameters and the computational burden. Specifically, we replace the  $3 \times 3$  convolution in the full pre-activation residual unit with two parallel  $3 \times 3$  atrous convolutions with different dilation rates and half the number of feature maps each. We then concatenate their outputs before the following  $1 \times 1$  convolution.

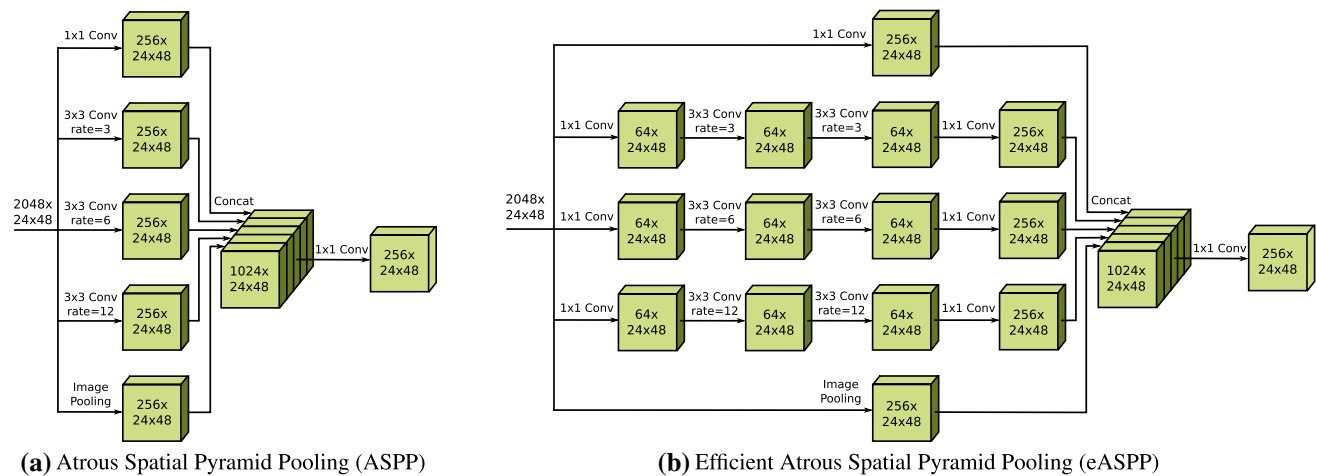
By concatenating their outputs, the network additionally learns to combine the feature maps of different scales. Now, by setting the dilation rate in one of the  $3 \times 3$  convolutional layers to one and another to a rate  $r \geq 2$ , we can preserve the original scale of the features within the block and simultaneously add a larger context. While, by varying the dilation rates in each of the parallel  $3 \times 3$  convolutions, we can enable the network to effectively learn multiscale representations at different stages of the network. The topology of the proposed multiscale residual units and the corresponding original residual units are shown in the legend in Fig. 3. The lower left two units show the original configuration, while the lower right two units show the proposed config-

uration. Figure 3 shows our entire encoder structure with the full pre-activation residual units and the multiscale residual units.

We incorporate the first multiscale residual unit with  $r_1 = 1, r_2 = 2$  before the third block at *res3d* (unit before the block where we remove the downsampling as mentioned earlier). Subsequently, we replace the units *res4c, res4d, res4e, res4f* with our proposed multiscale units with rates  $r_1 = 1$  in all the units and  $r_2 = 2, 4, 8, 16$  correspondingly. In addition, we replace the last three units of block four *res5a, res5b, res5c* with the multiscale units with increasing rates in both  $3 \times 3$  convolutions, as  $(r_1 = 2, r_2 = 4), (r_1 = 2, r_2 = 8), (r_1 = 2, r_2 = 16)$  correspondingly. We evaluate our proposed configuration in comparison to the multigrid method of DeepLab v3 (Chen et al. 2017) in Sect. 5.5.

### 3.2 Efficient Atrous Spatial Pyramid Pooling

In this section, we first describe the topology of the Atrous Spatial Pyramid Pooling (ASPP) module, followed by the structure of our proposed efficient Atrous Spatial Pyramid Pooling (eASPP). ASPP has become prevalent in most state-of-the-art architectures due to its ability to capture long range context and multiscale information. Inspired by spatial pyramid pooling (He et al. 2015c), the initially proposed ASPP in DeepLab v2 (Liang-Chieh et al. 2015) employs four parallel atrous convolutions with different dilation rates. Concatenating the outputs of multiple parallel atrous convolutions aggregates multi-scale context with different receptive field resolutions. However, as illustrated in the sub-



**Fig. 4** Depiction of the ASPP module from DeepLab v3 and our proposed efficient eASPP module. eASPP reduces the number of parameters by 87.87% and the number of FLOPs by 89.88%, while simultaneously achieving improved performance. Note that all the

sequent DeepLab v3 (Chen et al. 2017), applying extremely large dilation rates inhibits capturing long range context due to image boundary effects. Therefore, an improved version of ASPP was proposed (Chen et al. 2017) to add global context information by incorporating image-level features.

The resulting ASPP shown in Fig. 4a consists of five parallel branches: one  $1 \times 1$  convolution and three  $3 \times 3$  convolutions with different dilation rates. Additionally, image-level features are introduced by applying global average pooling on the input feature map, followed by a  $1 \times 1$  convolution and bilinear upsampling to yield an output with the same dimensions as the input feature map. All the convolutions have 256 filters and batch normalization layers to improve training. Finally, the resulting feature maps from each of the parallel branches are concatenated and passed through another  $1 \times 1$  convolution with batch normalization to yield 256 output filters. The ASPP module is appended after the last residual block of the encoder where the feature maps are of dimensions  $65 \times 65$  in the DeepLab v3 architecture (Chen et al. 2017), therefore dilation rates of 6, 12 and 18 were used in the parallel  $3 \times 3$  atrous convolution layers. However, as we use a smaller input image, the dimensions of the input feature map to the ASPP is  $24 \times 48$ , therefore, we reduce the dilation rates to 3, 6 and 12 in the  $3 \times 3$  atrous convolution layers respectively.

The biggest caveat of employing the ASPP is the extremely large amount of parameters and floating point operations per second (FLOPs) that it consumes. Each of the  $3 \times 3$  convolutions have 256 filters, which in total for the entire ASPP amounts to 15.53 M parameters and 34.58 B FLOPs which is prohibitively expensive. To address this problem, we propose an equivalent structure called eASPP that substantially reduces the computational complexity. Our proposed topol-

ogy is based on two principles: cascading atrous convolutions and the bottleneck structure. Cascading atrous convolutions effectively enlarges the receptive field as the latter atrous convolution takes the output of the former atrous convolution. The receptive field size  $F$  of an atrous convolution is computed as

$$F = (r - 1) \cdot (N - 1) + N, \quad (1)$$

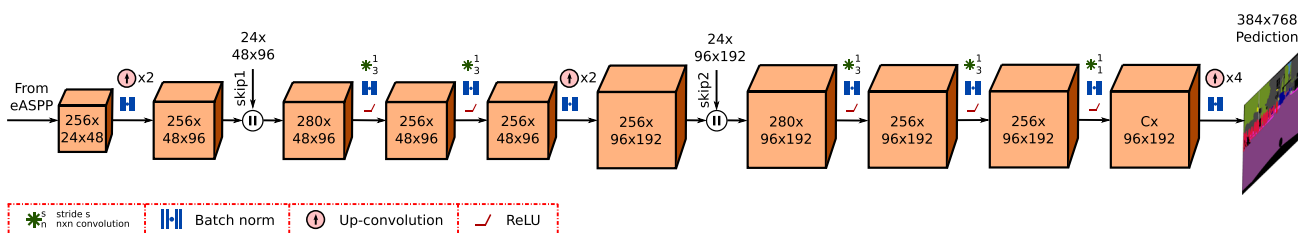
where  $r$  is the dilation rate of the atrous convolution and  $N$  is the filter size. When two atrous convolutions with the receptive field sizes as  $F_1$  and  $F_2$  are cascaded, the effective receptive field size is computed as

$$F_{eff} = F_1 + F_2 - 1. \quad (2)$$

For example, if two atrous convolutions with filter size  $F = 3$  and dilation  $r = 3$  are cascaded, then each of the convolutions individually has a receptive field size of 7, while the effective receptive field size of the second atrous convolution is 13. Moreover, cascading atrous convolutions enables denser sampling of pixels in comparison to parallel atrous convolution with a larger receptive field. Therefore, by using both parallel and cascaded atrous convolutions in the ASPP, we can efficiently aggregate dense multiscale features with very large receptive fields.

In order to reduce the number of parameters in the ASPP topology, we employ a bottleneck structure in the cascaded atrous convolution branches. The topology of our proposed eASPP shown in Fig. 4b consists of five parallel branches similar to ASPP but the branches with the  $3 \times 3$  atrous convolutions are replaced with our cascaded bottleneck branches. If  $c$  is the number of channels in the  $3 \times 3$  atrous convolution,





**Fig. 5** Our decoder consists of three upsampling stages that recover segmentation details using deconvolution layers and two skip refinement stages that fuse mid-level features from the encoder to improve the segmentation along object boundaries. Each skip refinement stage

consists of concatenation of mid-level features with the upsampled decoder feature maps, followed by two  $3 \times 3$  convolutions to improve the discriminability of the high-level features and the resolution of the refinement (Color figure online)

we add a  $1 \times 1$  convolution with  $c/4$  filters before the atrous convolution to squeeze only the most relevant information through the bottleneck. We then replace the  $3 \times 3$  atrous convolution with two cascaded  $3 \times 3$  atrous convolutions with  $c/4$  filters, followed by another  $1 \times 1$  convolution to restore the number of filters to  $c$ . The proposed eASPP only has 2.04 M parameters and consumes 3.62 B FLOPS which accounts to a reduction of 87.87% of parameters and 89.53% of FLOPS in comparison to the ASPP. We evaluate our proposed eASPP in comparison to ASPP in the ablation study presented in Sect. 5.5.2 and show that it achieves improved performance while being more than 10 times efficient in the number of parameters.

### 3.3 Decoder

The output of the eASPP in our network is 16-times downsampled with respect to the input image and therefore it has to be upsampled back to the full input resolution. In our previous work (Valada et al. 2017), we employed a simple decoder with two deconvolution layers and one skip refinement connection. Although the decoder was more effective in recovering the segmentation details in comparison to direct bilinear upsampling, it often produced disconnected segments while recovering the structure of thin objects such as poles and fences. In order to overcome this impediment, we propose a more effective decoder in this work.

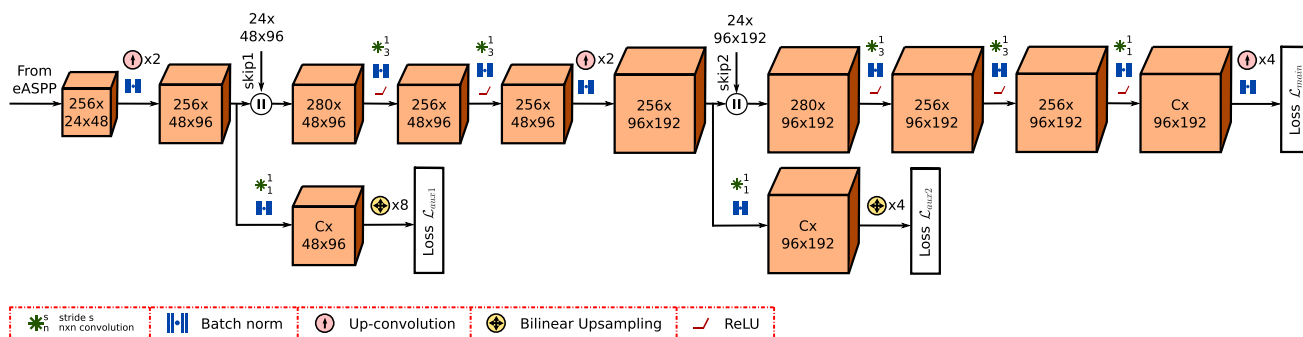
Our decoder shown in Fig. 5 consists of three stages. In the first stage, the output of the eASPP is upsampled by a factor of two using a deconvolution layer to obtain a coarse segmentation mask. The upsampled coarse mask is then passed through the second stage, where the feature maps are concatenated with the first skip refinement from *Res3d*. The skip refinement consists of a  $1 \times 1$  convolution layer to reduce the feature depth in order to not outweigh the encoder features. We experiment with varying number of feature channels in the skip refinement in the ablation study presented in Sect. 5.5.3. The concatenated feature maps are then passed through two  $3 \times 3$  convolutions to improve the

resolution of the refinement, followed by a deconvolution layer that again upsamples the feature maps by a factor of two. This upsampled output is fed to the last decoder stage which resembles the previous stage consisting of concatenation with the feature maps from the second skip refinement from *Res2c*, followed by two  $3 \times 3$  convolution layers. All the convolutional and deconvolutional layers until this stage have 256 feature channels, therefore the output from the two  $3 \times 3$  convolutions in the last stage is fed to a  $1 \times 1$  convolution layer to reduce the number of feature channels to the number of object categories  $C$ . This output is finally fed to the last deconvolution layer which upsamples the feature maps by a factor of four to recover the original input resolution.

### 3.4 Multiresolution Supervision

Deep networks often have difficulty in training due to the intrinsic instability associated with learning using gradient descent which leads to exploding or vanishing gradient problems. As our encoder is based on the residual learning framework, shortcut connections in each unit help propagating the gradient more effectively. Another technique that can be used to mitigate this problem to a certain extent is by initializing the layers with pretrained weights, however our proposed eASPP and decoder layers still have to be trained from scratch which could lead to optimization difficulties. Recent deep architectures have proposed employing an auxiliary loss in the middle of encoder network (Lee et al. 2015; Zhao et al. 2017), in addition to the main loss towards the end of the network. However, as shown in the ablation study presented in Sect. 5.5.1 this does not improve the performance of our network although it helps the optimization to converge faster.

Unlike previous approaches, in this work, we propose a multiresolution supervision strategy to both accelerate the training and improve the resolution of the segmentation. As described in the previous section, our decoder consists of three upsampling stages. We add two auxiliary loss branches at the end of the first and second stage after the deconvolution



**Fig. 6** Depiction of the two auxiliary softmax losses that we add before each skip refinement stage in the decoder in addition to the main softmax loss in the end of the decoder. The two auxiliary losses are weighed

for balancing the gradient flow through all the previous layers. While testing the auxiliary branches are removed and only the main stream as shown in Fig. 5 is used (Color figure online)

layer in addition to the main softmax loss  $\mathcal{L}_{main}$  at the end of the decoder as shown in Fig. 6. Each auxiliary loss branch decreases the feature channels to the number of category labels  $C$  using a  $1 \times 1$  convolution with batch normalization and upsamples the feature maps to the input resolution using bilinear upsampling. We only use simple bilinear upsampling which does not contain any weights instead of a deconvolution layer in the auxiliary loss branches as our aim is to force the main decoder stream to improve its discriminativeness at each upsampling resolution so that it embeds multiresolution information while learning to upsample. We weigh the two auxiliary losses  $\mathcal{L}_{aux1}$  and  $\mathcal{L}_{aux2}$  to balance the gradient flow through all the previous layers. While testing, the auxiliary loss branches are discarded and only the main decoder stream is used. We experiment with different loss weightings in the ablation study presented in Sects. 5.5.3 and in 5.5.1 we show that each of the auxiliary loss branches improves the segmentation performance in addition to speeding-up the training.

### 3.5 Network Compression

As we strive to design an efficient and compact semantic segmentation architecture that can be employed in resource constrained applications, we must ensure that the utilization of convolutional filters in our network is thoroughly optimized. Often, even the most compact networks have abundant neurons in deeper layers that do not significantly contribute to the overall performance of the model. Excessive convolutional filters not only increase the model size but also the inference time and the number of computing operations. These factors critically hinder the deployment of models in resource constrained real-world applications. Pruning of neural networks can be traced back to the 80s when LeCun et al. (1990) introduced a technique called Optimal Brain Damage for selectively pruning weights with a theoretically justified measure. Recently, several new techniques have been pro-

posed for pruning weight matrices (Wen et al. 2016; Anwar et al. 2017; Liu et al. 2017; Li et al. 2016) of convolutional layers as most of the computation during inference is consumed by them.

These approaches rank neurons based on their contribution and remove the low ranking neurons from the network, followed by fine-tuning of the pruned network. While the simplest neuron ranking method computes the  $\ell^1$ -norm of each convolutional filter (Li et al. 2016), more sophisticated techniques have recently been proposed (Anwar et al. 2017; Liu et al. 2017; Molchanov et al. 2017). Some of these approaches are based on sparsity based regularization of network parameters which additionally increases the computational overhead during training (Liu et al. 2017; Wen et al. 2016). Techniques have also been proposed for structured pruning of entire kernels with strided sparsity (Anwar et al. 2017) that demonstrate impressive results for pruning small networks. However, their applicability to complex networks that are to be evaluated on large validation sets has not been explored due its heavy computational processing. Moreover, until a year ago these techniques were only applied to simpler architectures such as VGG (Simonyan and Zisserman 2014) and AlexNet (Krizhevsky et al. 2012), as pruning complex deep architectures such as ResNets requires a holistic approach. Thus far, pruning of residual units has only been performed on convolutional layers that do not have an identity or shortcut connection as pruning them additionally requires pruning the added residual maps in the exact same configuration. Attempts to prune them in the same configuration have resulted in a significant drop in performance (Li et al. 2016). Therefore, often only the first and the second convolutional layers of a residual unit are pruned.

Our proposed AdapNet++ architecture has shortcut and skip connections both in the encoder as well the decoder. Therefore, in order to efficiently maximize the pruning of our network, we propose a holistic network-wide pruning technique that is invariant to the presence of skip or shortcut

connections. Our proposed technique first involves pruning all the convolutional layers of a residual unit, followed by masking out the pruned indices of the last convolutional layer of a residual unit with zeros before the addition of the residual maps from the shortcut connection. As masking is performed after the pruning, we efficiently reduce the parameters and computing operations in a holistic fashion, while optimally pruning all the convolutional layers and preserving the shortcut or skip connections. After each pruning iteration, we fine-tune the network to recover any loss in accuracy. We illustrate this strategy adopting a recently proposed greedy criteria-based oracle pruning technique that incorporates a novel ranking method based on a first order Taylor expansion of the network cost function (Molchanov et al. 2017). The pruning problem is framed as a combinatorial optimization problem such that when the weights  $B$  of the network are pruned, the change in cost value will be minimal.

$$\min_{\mathcal{W}'} |\mathcal{C}(\mathcal{T}|\mathcal{W}') - \mathcal{C}(\mathcal{T}|\mathcal{W})| \quad \text{s.t.} \quad \|\mathcal{W}'\|_0 \leq B, \quad (3)$$

where  $\mathcal{T}$  is the training set,  $\mathcal{W}$  is the network parameters and  $\mathcal{C}(\cdot)$  is the negative log-likelihood function. Based on Taylor expansion, the change in the loss function from removing a specific parameter can be approximated. Let  $h_i$  be the output feature maps produced by parameter  $i$  and  $h_i = \{z_0^1, z_0^2, \dots, z_L^i\}$ . The output  $h_i$  can be pruned by setting it to zero and the ranking can be given by

$$|\Delta\mathcal{C}(h_i)| = |\mathcal{C}(\mathcal{T}, h_i = 0) - \mathcal{C}(\mathcal{T}, h_i)|, \quad (4)$$

Approximating with Taylor expansion, we can write

$$\begin{aligned} \Theta_{TE}(h_i) &= |\Delta\mathcal{C}(h_i)| = \left| \mathcal{C}(\mathcal{T}, h_i) - \frac{\delta\mathcal{C}}{\delta h_i} h_i - \mathcal{C}(\mathcal{T}, h_i) \right| \\ &= \left| \frac{\delta\mathcal{C}}{\delta h_i} h_i \right|. \end{aligned} \quad (5)$$

$$\Theta_{TE}(z_l^{(k)}) = \left| \frac{1}{M} \sum_m \frac{\delta\mathcal{C}}{\delta z_{l,m}^{(k)}} z_{l,m}^{(k)} \right|, \quad (6)$$

where  $M$  is the length of the vectorized feature map. This ranking can be easily computed using the standard back-propagation computation as it requires the gradient of the cost function with respect to the activation and the product of the activation. Furthermore, in order to achieve adequate rescaling across layers, a layer-wise  $\ell^2$ -norm of the rankings is computed as

$$\hat{\Theta}(z_l^{(k)}) = \frac{\Theta(z_l^{(k)})}{\sqrt{\sum_j \Theta^2(z_l^{(j)})}}. \quad (7)$$

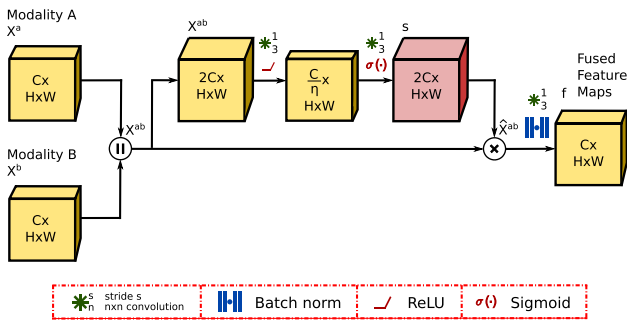
The entire pruning procedure can be summarized as follows: first the AdapNet++ network is trained until conver-

gence using the training protocol described in Sect. 5.1. Then the importance of the feature maps is evaluated using the aforementioned ranking method and subsequently the unimportant feature maps are removed. The pruned convolution layers that have shortcut connections are then masked at the indices where the unimportant feature maps are removed to maintain the shortcut connections. The network is then fine-tuned and the pruning process is reiterated until the desired trade-off between accuracy and the number of parameters has been achieved. We present results from pruning our AdapNet++ architecture in Sect. 5.4, where we perform pruning of both the convolutional and deconvolutional layers of our network in five stages by varying the threshold for the rankings. For each of these stages, we quantitatively evaluate the performance versus number of parameters trade-off obtained using our proposed pruning strategy in comparison to the standard approach.

### 4 Self-Supervised Model Adaptation

In this section, we describe our approach to multimodal fusion using our proposed self-supervised model adaptation (SSMA) framework. Our framework consists of three components: a modality-specific encoder as described in Sect. 3.1, a decoder built upon the topology described in Sect. 3.3 and our proposed SSMA block for adaptively recalibrating and fusing modality-specific feature maps. In the following, we first formulate the problem of semantic segmentation from multimodal data, followed by a detailed description of our proposed SSMA units and finally we describe the overall topology of our fusion architecture.

We represent the training set for multimodal semantic segmentation as  $\mathcal{T} = \{(I_n, K_n, M_n) \mid n = 1, \dots, N\}$ , where  $I_n = \{u_r \mid r = 1, \dots, \rho\}$  denotes the input frame from modality  $a$ ,  $K_n = \{k_r \mid r = 1, \dots, \rho\}$  denotes the corresponding input frame from modality  $b$  and the groundtruth label is given by  $M_n = \{m_r \mid r = 1, \dots, \rho\}$ , where  $m_r \in \{1, \dots, C\}$  is the set of semantic classes. The image  $I_n$  is only shown to the modality-specific encoder  $E_a$  and similarly, the corresponding image  $K_n$  from a complementary modality is only shown to the modality-specific encoder  $E_b$ . This enables each modality-specific encoder to specialize in a particular sub-space learning their own hierarchical representations individually. We assume that the input images  $I_n$  and  $K_n$ , as well as the label  $M_n$  have the same dimensions  $\rho = H \times W$  and that the pixels are drawn as *i.i.d.* samples following a categorical distribution. Let  $\theta$  be the network parameters consisting of weights and biases. Using the classification scores  $s_j$  at each pixel  $u_r$ , we obtain probabilities  $\mathbf{P} = (p_1, \dots, p_C)$  with the softmax function such that



**Fig. 7** The topology of our proposed SSMA unit that adaptively recalibrates and fuses modality-specific feature maps based on the inputs in order to exploit the more informative features from the modality-specific streams.  $\eta$  denotes the bottleneck compression rate (Color figure online)

$$p_j(u_r, \theta | I_n, K_n) = \sigma(s_j(u_r, \theta)) = \frac{\exp(s_j(u_r, \theta))}{\sum_k^C \exp(s_k(u_r, \theta))} \tag{8}$$

denotes the probability of pixel  $u_r$  being classified with label  $j$ . The optimal  $\theta$  is estimated by minimizing

$$\mathcal{L}_{seg}(\mathcal{T}, \theta) = - \sum_{n=1}^N \sum_{r=1}^{\rho} \sum_{j=1}^C \delta_{m_r, j} \log p_j(u_r, \theta | I_n, K_n), \tag{9}$$

for  $(I_n, K_n, M_n) \in \mathcal{T}$ , where  $\delta_{m_r, j}$  is the Kronecker delta.

### 4.1 SSMA Block

In order to adaptively recalibrate and fuse feature maps from modality-specific networks, we propose a novel architectural unit called the SSMA block. The goal of the SSMA block is to explicitly model the correlation between the two modality-specific feature maps before fusion so that the network can exploit the complementary features by learning to selectively emphasize more informative features from one modality, while suppressing the less informative features from the other. We construct the topology of the SSMA block in a fully-convolutional fashion which empowers the network with the ability to emphasize features from a modality-specific network for only certain spatial locations or object categories, while emphasizing features from the complementary modality for other locations or object categories. Moreover, the SSMA block dynamically recalibrates the feature maps based on the input scene context.

The structure of the SSMA block is shown in Fig. 7. Let  $\mathbf{X}^a \in \mathbb{R}^{C \times H \times W}$  and  $\mathbf{X}^b \in \mathbb{R}^{C \times H \times W}$  denote the modality-specific feature maps from modality A and modality B respectively, where  $C$  is the number of feature channels and  $H \times W$  is the spatial dimension. First, we concatenate the modality-specific feature maps  $\mathbf{X}^a$  and  $\mathbf{X}^b$  to yield

$\mathbf{X}^{ab} \in \mathbb{R}^{2 \cdot C \times H \times W}$ . We then employ a recalibration technique to adapt the concatenated feature maps before fusion. In order to achieve this, we first pass the concatenated feature map  $\mathbf{X}^{ab}$  through a bottleneck consisting of two  $3 \times 3$  convolutional layers for dimensionality reduction and to improve the representational capacity of the concatenated features. The first convolution has weights  $\mathcal{W}_1 \in \mathbb{R}^{\frac{1}{\eta} \cdot C \times H \times W}$  with a channel reduction ratio  $\eta$  and a non-linearity function  $\delta(\cdot)$ . We use ReLU for the non-linearity, similar to the other activations in the encoders and experiment with different reductions ratios in Sect. 5.10.2. Note that we omit the bias term to simplify the notation. The subsequent convolutional layer with weights  $\mathcal{W}_2 \in \mathbb{R}^{2 \cdot C \times H \times W}$  increases the dimensionality of the feature channels back to concatenation dimension  $2C$  and a sigmoid function  $\sigma(\cdot)$  scales the dynamic range of the activations to the  $[0, 1]$  interval. This can be represented as

$$\begin{aligned} \mathbf{s} &= F_{ssma}(\mathbf{X}^{ab}; \mathcal{W}) = \sigma(g(\mathbf{X}^{ab}; \mathcal{W})) \\ &= \sigma(\mathcal{W}_2 \delta(\mathcal{W}_1 \mathbf{X}^{ab})). \end{aligned} \tag{10}$$

The resulting output  $\mathbf{s}$  is used to recalibrate or emphasize/de-emphasize regions in  $\mathbf{X}^{ab}$  as

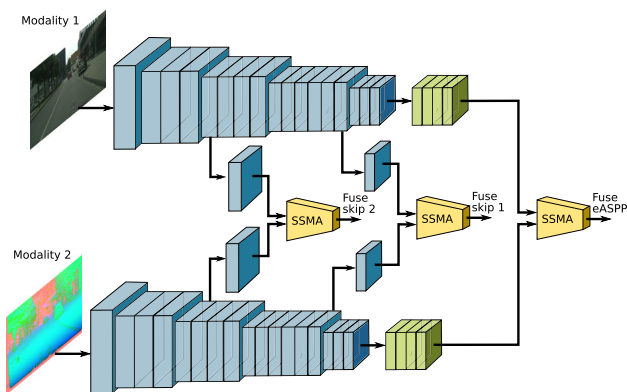
$$\hat{\mathbf{X}}^{ab} = F_{scale}(\mathbf{X}^{ab}; \mathbf{s}) = \mathbf{s} \circ \mathbf{X}^{ab}, \tag{11}$$

where  $F_{scale}(\mathbf{X}^{ab}, \mathbf{s})$  denotes Hadamard product of the feature maps  $\mathbf{X}^{ab}$  and the matrix of scalars  $\mathbf{s}$  such that each element  $x_{c,i,j}$  in  $\mathbf{X}^{ab}$  is multiplied with a corresponding activation  $s_{c,i,j}$  in  $\mathbf{s}$  with  $c \in \{1, 2, \dots, 2C\}$ ,  $i \in \{1, 2, \dots, H\}$  and  $j \in \{1, 2, \dots, W\}$ . The activations  $\mathbf{s}$  adapt to the concatenated input feature map  $\mathbf{X}^{ab}$ , enabling the network to weigh features element-wise spatially and across the channel depth based on the multimodal inputs  $I_n$  and  $K_n$ . With new multimodal inputs, the network dynamically weighs and reweighs the feature maps in order to optimally combine complementary features. Finally, the recalibrated feature maps  $\hat{\mathbf{X}}^{ab}$  are passed through a  $3 \times 3$  convolution with weights  $\mathcal{W}_3 \in \mathbb{R}^{C \times H \times W}$  and a batch normalization layer to reduce the feature channel depth and yield the fused output  $\mathbf{f}$  as

$$\mathbf{f} = F_{fused}(\hat{\mathbf{X}}^{ab}; \mathcal{W}) = g(\hat{\mathbf{X}}^{ab}; \mathcal{W}) = \mathcal{W}_3 \hat{\mathbf{X}}^{ab}. \tag{12}$$

As described in the following section, we employ our proposed SSMA block to fuse modality-specific feature maps both at intermediate stages of the network and towards the end of the encoder. Although we utilize a bottleneck structure to conserve the number of parameters consumed, further reduction in the parameters can be achieved by replacing the  $3 \times 3$  convolution layers with  $1 \times 1$  convolutions, which yields comparable performance. We also remark that the SSMA





**Fig. 8** Topology of our Adapnet++ encoder for multimodal fusion. The encoder employs a late fusion technique to fuse feature maps from modality-specific streams using our proposed SSMA block. The SSMA block is employed to fuse the latent features from the eASPP as well as the feature maps from the skip refinements (Color figure online)

blocks can be used for multimodal fusion in other tasks such as scene classification as shown in Sect. 5.9.

### 4.2 Fusion Architecture

We propose a framework for multimodal semantic segmentation using a modified version of our AdapNet++ architecture and the proposed SSMA blocks. For simplicity, we consider the fusion of two modalities, but the framework can be easily extended to arbitrary number of modalities. The encoder of our framework shown in Fig. 8 contains two streams, where each stream is based on the encoder topology described in Sect. 3.1. Each encoder stream is modality-specific and specializes in a particular sub-space. In order to fuse the feature maps from both streams, we adopt a combination of mid-level and late fusion strategy in which we fuse the latent representations of both encoders using the SSMA block and pass the fused feature map to the first decoder stage. We denote this as latent SSMA fusion as it takes the output of the eASPP from each modality-specific encoder as input. We set the reduction ratio  $\eta = 16$  in the latent SSMA. As the AdapNet++ architecture contains skip connections for high-resolution refinement, we employ an SSMA block at each skip refinement stage after the  $1 \times 1$  convolution as shown in Fig. 8. As the  $1 \times 1$  convolutions reduce the feature channel depth to 24, we only use a reduction ratio  $\eta = 6$  in the two skip SSMA blocks as identified from the ablation experiments presented in Sect. 5.10.2.

In order to upsample the fused predictions, we build upon our decoder described in Sect. 3.3. The main stream of our decoder resembles the topology of the decoder in our AdapNet++ architecture consisting of three upsampling stages. The output of the latent SSMA block is fed to the first upsampling stage of the decoder. Following the AdapNet++ topology, the outputs of the skip SSMA blocks would be

concatenated into the decoder at the second and third upsampling stages (*skip1* after the first deconvolution and *skip2* after the second deconvolution). However, we find that concatenating the fused mid-level features into the decoder does not substantially improve the resolution of the segmentation, as much as in the unimodal AdapNet++ architecture. We hypothesise that directly concatenating the fused mid-level features and fused high-level features causes a feature localization mismatch as each SSMA block adaptively recalibrates at different stages of the network where the resolution of the feature maps and channel depth differ by one half of their dimensions. Moreover, training the fusion network end-to-end from scratch also contributes to this problem as without initializing the encoders with modality-specific pre-trained weights, concatenating the uninitialized mid-level fused encoder feature maps into the decoder does not yield any performance gains, rather it hampers the convergence.

With the goal of mitigating this problem, we propose two strategies. In order to facilitate better fusion, we adopt a multi-stage training protocol where we first initialize each encoder in the fusion architecture with pre-trained weights from the unimodal AdapNet++ model. We describe this procedure in Sect. 5.1.2. Secondly, we propose a mechanism to better correlate the mid and high-level fused features before concatenation in the decoder. We propose to weigh the fused mid-level skip features with the spatially aggregated statistics of the high-level decoder features before the concatenation. Following the notation convention, we define  $\mathbf{D} \in \mathbb{R}^{C \times H \times W}$  as the high-level decoder feature map before the skip concatenation stage. A feature statistic  $\mathbf{s} \in \mathbb{R}^C$  is produced by projecting  $\mathbf{D}$  along the spatial dimensions  $H \times W$  using a global average pooling layer as

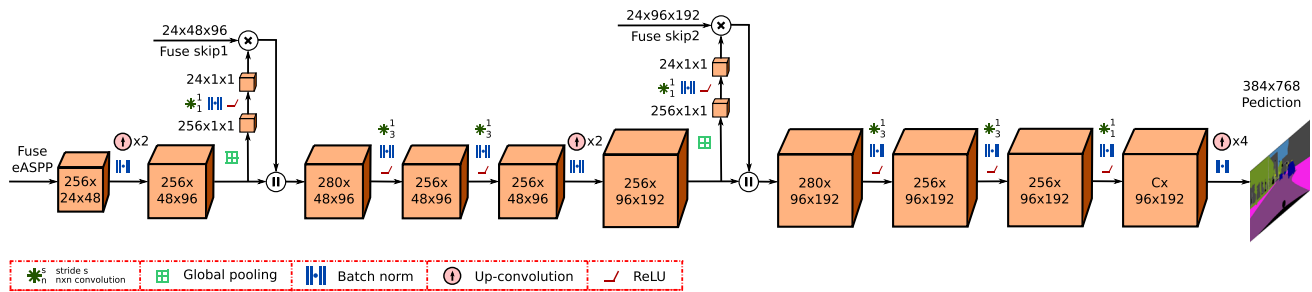
$$s_c = F_{shrink}(d_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W d_c(i, j), \tag{13}$$

where  $s_c$  represents a statistic or a local descriptor of the  $c^{th}$  element of  $\mathbf{D}$ . We then reduce the number of feature channels in  $\mathbf{s}$  using a  $1 \times 1$  convolution layer with weights  $\mathcal{W}_4 \in \mathbb{R}^{C \times H \times W}$ , batch normalization and an ReLU activation function  $\delta$  to match the channels of the fused mid-level feature map  $\mathbf{f}$ , where  $\mathbf{f}$  is computed as shown in Eq. (12). We can represent resulting output as

$$\mathbf{z} = F_{reduce}(\mathbf{s}; \mathcal{W}) = \delta(\mathcal{W}_4 \mathbf{s}). \tag{14}$$

Finally, we weigh the fused mid-level feature map  $\mathbf{f}$  with the reduced aggregated descriptors  $\mathbf{z}$  using channel-wise multiplication as

$$\hat{\mathbf{f}} = F_{loc}(\mathbf{f}_c; \mathbf{z}_c) = (z_1 \mathbf{f}_1, z_2 \mathbf{f}_2, \dots, z_c \mathbf{f}_c). \tag{15}$$



**Fig. 9** Topology of the modified AdapNet++ decoder used for multi-modal fusion. We propose a mechanism to better correlate the fused mid-level skip refinement features with the high-level decoder fea-

ture before integrating into the decoder. The correlation mechanism is depicted following the fuse skip connections (Color figure online)

As shown in Fig. 9, we employ the aforementioned mechanism to the fused feature maps from skip1 SSMA as well as skip2 SSMA and concatenate their outputs with the decoder feature maps at the second and third upsampling stages respectively. We find that this mechanism guides the fusion of mid-level skip refinement features with the high-level decoder feature more effectively than direct concatenation and yields a notable improvement in the resolution of the segmentation output.

## 5 Experimental Results

In this section, we first describe the datasets that we benchmark on, followed by comprehensive quantitative results for unimodal segmentation using our proposed AdapNet++ architecture in Sect. 5.3 and the results for model compression in Sect. 5.4. We then present detailed ablation studies that describe our architectural decisions in Sect. 5.5, followed by the qualitative unimodal segmentation results in Sect. 5.6. We present the multimodal fusion benchmarking experiments with the various modalities contained in the datasets in Sect. 5.7 and the ablation study on our multimodal fusion architecture in Sect. 5.10. We finally present the qualitative multimodal segmentation results in Sect. 5.11 and in challenging perceptual conditions in Sect. 5.12.

All our models were implemented using the TensorFlow (Abadi et al. 2015) deep learning library and the experiments were carried out on a system with an Intel Xeon E5 with 2.4 GHz and an NVIDIA TITAN X GPU. We primarily use the standard Jaccard Index, also known as the intersection-over-union (IoU) metric to quantify the performance. The IoU for each object class is computed as  $\text{IoU} = \text{TP} / (\text{TP} + \text{FP} + \text{FN})$ , where TP, FP and FN correspond to true positives, false positives and false negatives respectively. We report the mean intersection-over-union (mIoU) metric for all the models and also the pixel-wise accuracy (Acc), average precision (AP), global intersection-over-union (gIoU) metric, false positive rate (FPR), false negative rate (FNR) in the detailed analysis. All the metrics are computed as

defined in the PASCAL VOC challenge (Everingham et al. 2015) and additionally, the gIoU metric is computed as  $\text{gIoU} = \sum_C \text{TP}_C / \sum_C (\text{TP}_C + \text{FP}_C + \text{FN}_C)$ , where C is the number of object categories. The implementations of our proposed architectures are publicly available at <https://github.com/DeepSceneSeg> and a live demo can be viewed at <http://deepsceen.cs.uni-freiburg.de>.

### 5.1 Training Protocol

In this section, we first describe the procedure that we employ for training our proposed AdapNet++ architecture, followed by the protocol for training the SSMA fusion scheme. We then detail the various data augmentations that we perform on the training set.

#### 5.1.1 AdapNet++ Training

We train our network with an input image of resolution  $768 \times 384$  pixels, therefore we employ bilinear interpolation for resizing the RGB images and the nearest-neighbor interpolation for the other modalities as well as the groundtruth labels. We initialize the encoder section of the network with weights pre-trained on the ImageNet dataset (Deng et al. 2009), while we use the He initialization (He et al. 2015b) for the other convolutional and deconvolutional layers. We use the Adam solver for optimization with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\epsilon = 10^{-10}$ . We train our model for 150K iterations using an initial learning rate of  $\lambda_0 = 10^{-3}$  with a mini-batch size of 8 and a dropout probability of 0.5. We use the cross-entropy loss function and set the weights  $\lambda_1 = 0.6$  and  $\lambda_2 = 0.5$  to balance the auxiliary losses. The final loss function can be given as  $\mathcal{L} = \mathcal{L}_{main} + \lambda_1 \mathcal{L}_{aux1} + \lambda_2 \mathcal{L}_{aux2}$ .

#### 5.1.2 SSMA Training

We employ a multi-stage procedure for training the multimodal models using our proposed SSMA fusion scheme. We first train each modality-specific Adapnet++ model individually using the training procedure described in Sect. 5.1.1.

In the second stage, we leverage transfer learning to train the joint fusion model in the SSMA framework by initializing only the encoders with the weights from the individual modality-specific encoders trained in the previous stage. We then set the learning rate of the encoder layers to  $\lambda_0 = 10^{-4}$  and the decoder layers to  $\lambda_0 = 10^{-3}$ , and train the fusion model with a mini-batch of 7 for a maximum of 100K iterations. This enables the SSMA blocks to learn the optimal combination of multimodal feature maps from the well trained encoders, while slowly adapting the encoder weights to improve the fusion. In the final stage, we fix the learning rate of the encoder layers to  $\lambda_0 = 0$  while only training the decoder and the SSMA blocks with a learning rate of  $\lambda_0 = 10^{-5}$  and a mini-batch size of 12 for 50K iterations. This enables us to train the network with a larger batch size, while focusing more on the upsampling stages to yield the high-resolution segmentation output.

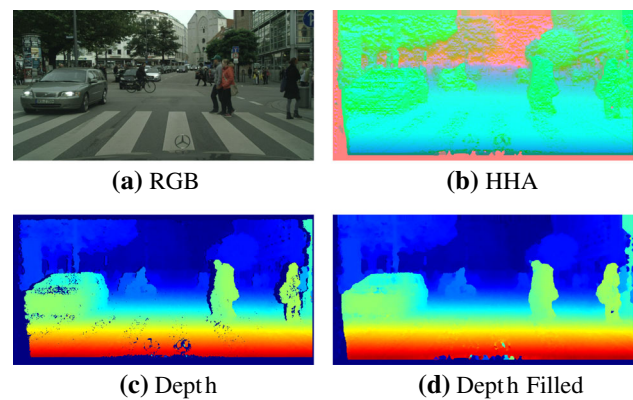
### 5.1.3 Data Augmentation

The training of deep networks can be significantly improved by expanding the dataset to introduce more variability. In order to achieve this, we apply a series of augmentation strategies randomly on the input data while training. The augmentations that we apply include rotation ( $-13^\circ$  to  $13^\circ$ ), skewing (0.05–0.10), scaling (0.5–2.0), vignetting (210–300), cropping (0.8–0.9), brightness modulation ( $-40$  to  $40$ ), contrast modulation (0.5–1.5) and flipping.

## 5.2 Datasets

We evaluate our proposed AdapNet++ architecture on five publicly available diverse scene understanding benchmarks ranging from urban driving scenarios to unstructured forested scenes and cluttered indoor environments. The datasets were particularly chosen based on the criteria of containing scenes with challenging perceptual conditions including rain, snow, fog, night-time, glare, motion blur and other seasonal appearance changes. Each of the datasets contain multiple modalities that we utilize for benchmarking our fusion approach. We briefly describe the datasets and their constituting semantic categories in this section.

**Cityscapes** The Cityscapes dataset (Cordts et al. 2016) is one of the largest labeled RGB-D dataset for urban scene understanding. Being one of the standard benchmarks, it is highly challenging as it contains images of complex urban scenes, collected from over 50 cities during varying seasons, lighting and weather conditions. The images were captured using a automotive-grade 22 cm baseline stereo camera at a resolution of  $2048 \times 1024$  pixels. The dataset contains 5000 finely annotated images, of which 2875 are provided for training, 500 are provided for validation and 1525 are used

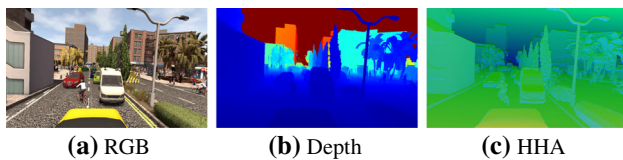


**Fig. 10** Example image from the Cityscapes dataset showing a complex urban scene with many dynamic objects and the corresponding depth map representations (Color figure online)

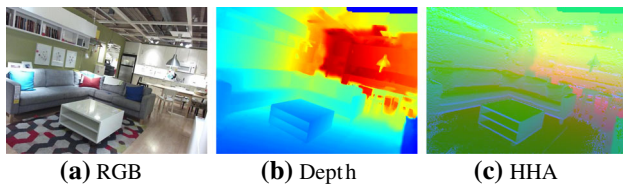
for testing. As a supplementary training set, 20,000 coarse annotations are also provided. The testing images are not publicly released, they are used by the evaluation server for benchmarking on 19 semantic object categories. We report results on the full 19 class label set for both the validation and test sets. Additionally, in order to facilitate comparison with previous fusion approaches we also report results on the reduced 11 class label set consisting of: *sky, building, road, sidewalk, fence, vegetation, pole, car/truck/bus, traffic sign, person, rider/bicycle/motorbike* and *background*.

In our previous work (Valada et al. 2017), we directly used the colorized depth image as input to our network. We converted the stereo disparity map to a three-channel colorized depth image by normalizing and applying the standard jet color map Fig. 10a, c show an example image and the corresponding colorized depth map from the dataset. However, as seen in the figure, the depth maps have considerable amount of noise and missing depth values due to occlusion, which are undesirable especially when utilizing depth maps as an input modality for pixel-wise segmentation. Therefore, in this work, we employ a recently proposed state-of-the-art fast depth completion technique (Ku et al. 2018) to fill any holes that may be present. The resulting filled depth map is shown in Fig. 10d. The depth completion algorithm can easily be incorporated into our pipeline as a preprocessing step as it only requires 11ms while running on the CPU and it can be further parallelized using a GPU implementation. Additionally, Gupta et al. (2014) proposed an alternate representation of the depth map known as the HHA encoding to enable DCNNs to learn more effectively. The authors demonstrate that the HHA representation encodes properties of geocentric pose that emphasizes on complementary discontinuities in the image which are extremely hard for the network to learn, especially from limited training data. This representation also yields a three-channel image consisting of: horizontal disparity, height above ground, and the angle between the local surface normal of a pixel and the





**Fig. 11** Example image from the Synthia dataset showing an outdoor urban scene and the corresponding depth map representations (Color figure online)

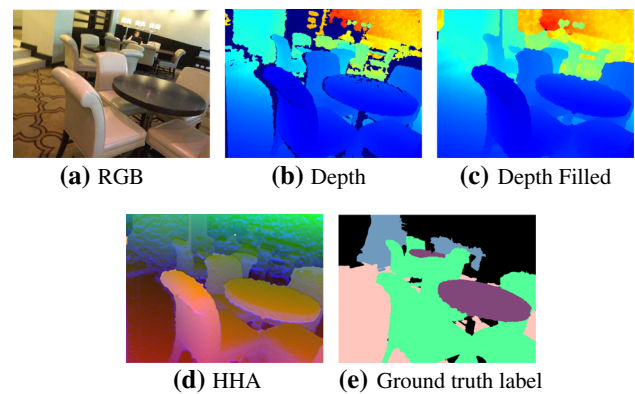


**Fig. 12** Example image from the SUN RGB-D dataset showing an indoor scene and the corresponding depth map representations (Color figure online)

inferred gravity direction. The resulting channels are then linearly scaled and mapped to the 0 to 255 range. However, it is still unclear if this representation enables the network to learn features complementary to that learned from visual RGB images as different works show contradicting results (Hazirbas et al. 2016; Gupta et al. 2014; Eitel et al. 2015). In this paper, we perform in-depth experiments with both the jet colored and the HHA encoded depth map on a larger and more challenging dataset than previous works to investigate the utility of these encodings.

**Synthia** The Synthia dataset (Ros et al. 2016) is a large-scale urban outdoor dataset that contains photo realistic images and depth data rendered from a virtual city built using the Unity engine. An example image and the corresponding modalities from this dataset is shown in Fig. 11. It consists of several annotated label sets. In this work, we use the Synthia-Rand-Cityscapes and the video sequences which have images of resolution  $1280 \times 760$  with a  $100^\circ$  horizontal field of view. This dataset is of particular interest for benchmarking the fusion approaches as it contains diverse traffic situations under different weather conditions. Synthia-Rand-Cityscapes consists of 9000 images and the sequences contain 8000 images with groundtruth labels for 12 classes. The categories of object labels are the same as the aforementioned Cityscapes label set.

**SUN RGB-D** The SUN RGB-D dataset (Song et al. 2015) is one of the most challenging indoor scene understanding benchmarks to date. It contains 10,335 RGB-D images that were captured with four different types of RGB-D cameras (Kinect V1, Kinect V2, Xtion and RealSense) with different resolutions and fields of view. This benchmark also combines several other datasets including 1449 images from the NYU Depth v2 (Silberman et al. 2012), 554 images from the Berkeley B3DO (Janoch et al. 2013) and 3389 images



**Fig. 13** Example image from the ScanNet dataset showing a complex indoor scene, the corresponding depth map representations and the groundtruth semantic segmentation mask (Color figure online)

from the SUN3D (Xiao et al. 2013). An example image and the corresponding modalities from this dataset is shown in Fig. 12. We use the original train-val split consisting of 5285 images for training and 5050 images for testing. We use the refined in-painted depth images from the dataset that were processed using a multi-view fusion technique. However, some refined depth images still have missing depth values at distances larger than a few meters. Therefore, as mentioned in previous works (Hazirbas et al. 2016), we exclude the 587 training images that were captured using the RealSense RGB-D camera as they contain a significant amount of invalid depth measurements that are further intensified due to the in-painting process.

This dataset provides pixel-level semantic annotations for 37 categories, namely: *wall, floor, cabinet, bed, chair, sofa, table, door, window, bookshelf, picture, counter, blinds, desk, shelves, curtain, dresser, pillow, mirror, floor mat, clothes, ceiling, books, fridge, tv, paper, towel, shower curtain, box, whiteboard, person, night stand, toilet, sink, lamp, bathtub and bag*. Although we benchmark on all the object categories, 16 out of the 37 classes are rarely present in the images and about 0.25% of the pixels are not assigned to any of the classes, making it extremely unbalanced. Moreover, as each scene contains many different types of objects, they are often partially occluded and may appear completely different in the test images.

**ScanNet** The ScanNet RGB-D video dataset (Dai et al. 2017) is a recently introduced large-scale indoor scene understanding benchmark. It contains 2.5M RGB-D images accounting to 1512 scans acquired in 707 distinct spaces. The data was collected using an iPad Air2 mounted with a depth camera similar to the Microsoft Kinect v1. Both the iPad camera and the depth camera were hardware synchronized and frames were captured at 30Hz. The RGB images were captured at a resolution of  $1296 \times 968$  pixels and the depth frames were captured at  $640 \times 480$  pixels. The semantic segmenta-

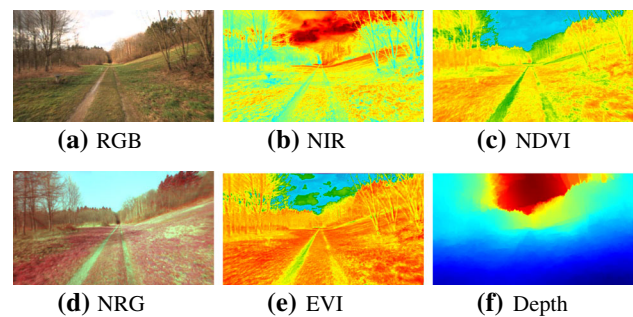


tion benchmark contains 16,506 labelled training images and 2537 testing images. From the example depth image shown in Fig. 13b, we can see that there are a number of missing depth values at the object boundaries and at large distances. Therefore, similar to the preprocessing that we perform on the cityscapes dataset, we use a fast depth completion technique (Ku et al. 2018) to fill the holes. The corresponding filled depth image is shown in Fig. 13c. We also compute the HHA encoding for the depth maps and use them as an additional modality in our experiments.

The dataset provides pixel-level semantic annotations for 21 object categories, namely: *wall, floor, chair, table, desk, bed, bookshelf, sofa, sink, bathtub, toilet, curtain, counter, door, window, shower curtain, refrigerator, picture, cabinet, other furniture* and *void*. Similar to the SUN RGB-D dataset, many object classes are rarely present making the dataset unbalanced. Moreover, the annotations at the object boundaries are often irregular and parts of objects at large distances are unlabelled as shown in Fig. 13e. These factors make the task even more challenging on this dataset.

**Freiburg Forest** In our previous work (Valada et al. 2016b), we introduced the Freiburg Multispectral Segmentation benchmark, which is a first-of-a-kind dataset of unstructured forested environments. Unlike urban and indoor scenes which are highly structured with rigid objects that have distinct geometric properties, objects in unstructured forested environments are extremely diverse and moreover, their appearance completely changes from month to month due to seasonal variations. The primary motivation for the introduction of this dataset is to enable robots to discern obstacles that can be driven over such as tall grass and bushes to obstacles that should be avoided such as tall trees and boulders. Therefore, we proposed to exploit the presence of chlorophyll in these objects which can be detected in the Near-InfraRed (NIR) wavelength. NIR images provide a high fidelity description on the presence of vegetation in the scene and as demonstrated in our previous work (Valada et al. 2017), it enhances border accuracy for segmentation.

The dataset was collected over an extended period of time using our Viona autonomous robot equipped with a Bumblebee2 camera to capture stereo images and a modified camera with the NIR-cut filter replaced with a Wratten 25A filter for capturing the NIR wavelength in the blue and green channels. The dataset contains over 15,000 images that were sub-sampled at 1 Hz, corresponding to traversing over 4.7 km each day. In order to extract consistent spatial and global vegetation information we computed vegetation indices such as Normalized Difference Vegetation Index (NDVI) and Enhanced Vegetation Index (EVI) using the approach presented by Huete et al. (1999). NDVI is resistant to noise caused due to changing sun angles, topography and shadows but is susceptible to error due to variable atmospheric



**Fig. 14** Example image from the Freiburg Forest dataset showing the various spectra and modalities: Near-InfraRed (NIR), Normalized Difference Vegetation Index (NDVI), Near-InfraRed-Red-Green (NRG), Enhanced Vegetation Index (EVI) and Depth (Color figure online)

and canopy background conditions (Huete et al. 1999). EVI was proposed to compensate for these defects with improved sensitivity to high biomass regions and improved detection though decoupling of canopy background signal and reduction in atmospheric influences. Figure 14 shows an example image from the dataset and the corresponding modalities. The dataset contains hand-annotated segmentation groundtruth for six classes: *sky, trail, grass, vegetation, obstacle* and *void*. We use the original train and test splits provided by the dataset.

### 5.3 AdapNet++ Benchmarking

In this section, we report results comparing the performance of our proposed AdapNet++ architecture against several well adopted state-of-the-art models including DeepLab v3 (Chen et al. 2017), ParseNet (Liu et al. 2015), FCN-8s (Long et al. 2015), SegNet (Badrinarayanan et al. 2015), FastNet (Oliveira et al. 2016), DeepLab v2 (Chen et al. 2016), DeconvNet (Noh et al. 2015) and Adapnet (Valada et al. 2017). We use the official implementations of these architectures that are publicly released by the authors to train on the input image resolution of  $768 \times 384$  pixels. For the Cityscapes and ScanNet benchmarking results reported in Tables 2 and 6, we report results directly from the official benchmark leaderboard. For each of the datasets, we report the mIoU score, as well as the per-class IoU score. In order to have a fair comparison, we also evaluate the models at the same input resolution using the same evaluation settings. We do not apply multiscale inputs or left–right flips during testing as these techniques require each crop of each image to be evaluated several times which significantly increases the computational complexity and runtime (Note: We do not use crops for testing, we evaluate on the full image in a single forward-pass). Moreover, these techniques do not improve the performance of the model in real-time applications. However, we show the potential gains that can be obtained in the evaluation metric utilizing these techniques and with a

**Table 1** Performance comparison of AdapNet++ with baseline models on the Cityscapes validation set with 11 semantic class labels (input image dim: 768 × 384) (Color table online)

Network	Sky ■	Building ■	Road ■	Sidewalk ■	Fence ■	Vegetation ■	Pole ■	Car ■	Sign ■	Person ■	Cyclist ■	mIoU (%)
FCN-8s	76.51	83.97	93.82	67.67	24.91	86.38	31.71	84.80	50.92	59.89	59.11	59.97
SegNet	73.74	79.29	92.70	59.88	13.63	81.89	26.18	78.83	31.44	45.03	43.46	52.17
FastNet	77.69	86.25	94.97	72.99	31.02	88.06	38.34	88.42	52.34	61.76	61.83	68.52
ParseNet	77.57	86.81	95.27	74.02	33.31	87.37	38.24	88.99	53.34	63.25	63.87	69.28
DeconvNet	89.38	83.08	95.26	68.07	27.58	85.80	34.20	85.01	27.62	45.11	41.11	62.02
DeepLab v2	74.28	81.66	90.86	63.3	26.29	84.33	27.96	86.24	44.79	58.89	60.92	63.59
AdapNet	92.45	89.98	97.43	81.43	49.93	91.44	53.43	92.23	65.32	69.86	69.62	77.56
DeepLab v3	92.82	89.02	96.74	78.13	41.00	90.81	49.74	91.02	64.48	66.52	66.98	75.21
AdapNet++ (ours)	<b>94.18</b>	<b>91.49</b>	<b>97.93</b>	<b>84.40</b>	<b>54.98</b>	<b>92.09</b>	<b>58.85</b>	<b>93.86</b>	<b>72.61</b>	<b>75.52</b>	<b>72.90</b>	<b>80.80</b>

Note that no left–right flips or multiscale testing is performed. Corresponding multimodal results are reported in Table 17  
 Bold numbers indicate the best performance among the compared methods

higher resolution input image in the ablation study presented in Sect. 5.5.6. Additionally, we report results with full resolution evaluation on the test set of the datasets when available, namely for Cityscapes and ScanNet.

Table 1 shows the results on the 11 class Cityscapes validation set. AdapNet++ outperforms all the baselines in each individual object category as well in the mIoU score. AdapNet++ outperforms the highest baseline by a margin of 3.24%. Analyzing the individual class IoU scores, we can see that AdapNet++ yields the highest improvement in object categories that contain thin structures such as *poles* for which it gives a large improvement of 5.42%, a similar improvement of 5.05% for *fences* and the highest improvement for 7.29% for *signs*. Most architectures struggle to recover the structure of thin objects due to downsampling by pooling and striding in the network which causes such information to be lost. However, these results show that AdapNet++ efficiently recovers the structure of such objects by learning multiscale features at several stages of the encoder using the proposed multiscale residual units and the eASPP. We further show the improvement in performance due to the incorporation of the multiscale residual units and the eASPP in the ablation study presented in Sect. 5.5.1. In driving scenarios, information of objects such as *pedestrians* and *cyclists* can also be lost when they appear at far away distances. A large improvement can also be seen in categories such as *person* in which AdapNet++ achieves an improvement of 5.66%. The improvement in larger object categories such as *cars* and *vegetation* can be attributed to the new decoder which improves the segmentation performance near object boundaries. This is more evident in the qualitative results presented in Sect. 5.11. Note that the colors shown below the object category names serve as a legend for the qualitative results.

We also report results on the full 19 class Cityscapes validation and test sets in Table 2. We compare against the top six

published models on the leaderboard, namely, PSPNet (Zhao et al. 2017), DeepLab v3 (Chen et al. 2017), Mapillary (Bulò et al. 2018), DeepLab v3+ (Chen et al. 2018b), DPC (Chen et al. 2018a), and DRN (Zhuang et al. 2018). The results of the competing methods reported in this table are directly taken from the benchmark leaderboard for the test set and from the corresponding manuscripts of the methods for the validation set. We trained our models on 768 × 768 crops from the full image resolution for benchmarking on the leaderboard. Our AdapNet++ model with a much smaller network backbone achieves a comparable performance as other top performing models on the leaderboard. Moreover, our network is the most efficient architecture in terms of both the number of parameters that it consumes as well as the inference time compared to other networks on the entire first page of the Cityscapes leaderboard.

We benchmark on the Synthia dataset largely due to the variety of seasons and adverse perceptual conditions where the improvement due to multimodal fusion can be seen. However, even for baseline comparison shown in Table 3, it can be seen that AdapNet++ outperforms all the baselines, both in the overall mIoU score as well as in the score of the individual object categories. It achieves an overall improvement of 3.87% and a similar observation can be made in the improvement of scores for thin structures, reinforcing the utility of our proposed multiscale feature learning configuration. The largest improvement of 13.14% was obtained for the *sign* class, followed by an improvement of 7.8% for the *pole* class. In addition a significant improvement of 5.42% can also be seen for the *cyclist* class.

Compared to outdoor driving datasets, indoor benchmarks such as SUN RGB-D and ScanNet pose a different challenge. Indoor datasets have vast amounts of object categories in various different configurations and images captured from many different view points compared to driving scenarios where the

**Table 2** Benchmarking results on the Cityscapes dataset with full resolution evaluation on 19 semantic class labels

Network	Backbone	mIoU (%)		Params. (M)	Time (ms)
		Val	Test		
PSPNet	ResNet-101	80.91	81.19	56.27	172.42
DeepLab v3	ResNet-101	79.30	81.34	58.16	79.90
Mapillary	WideResNet-38	78.31	82.03	135.86	214.46
DeepLab v3+	Modified Xception	79.55	82.14	43.48	127.97
DPC	Modified Xception	80.85	82.66	41.82	144.41
DRN	WideResNet-38	79.69	82.82	129.16	1259.67
AdapNet++ (ours)	ResNet-50	<b>81.24</b>	<b>81.34</b>	<b>30.20</b>	72.92
SSMA (ours)	ResNet-50	<b>82.19</b>	<b>82.31</b>	<b>56.44</b>	101.95

Only the eight top performing published models in the leaderboard are listed in this table. The inference time is reported for an input image resolution of  $768 \times 384$  pixels and it was computed on an NVIDIA TITAN X (PASCAL) GPU using the official implementation of each method  
Bold numbers indicate the best performance among the compared methods

**Table 3** Performance comparison of AdapNet++ with baseline models on the Synthia validation set (input image dim:  $768 \times 384$ ) (Color table online)

Network	Sky	Building	Road	Sidewalk	Fence	Vegetation	Pole	Car	Sign	Person	Cyclist	mIoU (%)
FCN-8s	92.36	91.92	88.94	86.46	48.22	77.41	36.02	82.63	30.37	57.10	46.84	67.11
SegNet	91.90	87.19	83.72	80.94	50.02	71.63	26.12	71.31	1.01	52.34	32.64	58.98
FastNet	92.21	92.41	91.85	89.89	56.64	78.59	51.17	84.75	32.03	69.87	55.65	72.28
ParseNet	93.80	93.09	91.05	88.98	53.22	79.48	46.15	85.37	36.00	63.30	50.82	71.02
DeconvNet	95.88	93.83	92.85	90.79	66.40	81.04	48.23	84.65	0.00	69.46	52.79	70.54
DeepLab v2	94.07	93.34	88.07	88.93	55.57	80.22	45.97	85.87	38.73	64.40	52.54	71.61
AdapNet	96.95	95.88	95.60	94.46	76.30	86.59	67.14	92.20	58.85	80.18	66.89	82.83
DeepLab v3	95.30	92.75	93.58	91.56	73.37	80.71	55.83	88.09	44.17	75.65	60.15	77.38
AdapNet++ (ours)	<b>97.77</b>	<b>96.98</b>	<b>96.60</b>	<b>95.70</b>	<b>79.87</b>	<b>89.63</b>	<b>74.94</b>	<b>94.22</b>	<b>71.99</b>	<b>83.64</b>	<b>72.31</b>	<b>86.70</b>

Note that no left–right flips or multiscale testing is performed. Corresponding multimodal results are reported in Table 18  
Bold numbers indicate the best performance among the compared methods

camera is always parallel to the ground with similar viewpoints from the perspective of the vehicle driving on the road. Moreover, indoor scenes are often extremely cluttered which causes occlusions, in addition to the irregular frequency distribution of the object classes that make the problem even harder. Due to these factors SUN RGB-D is considered one of the hardest datasets to benchmark on. Despite these factors, as shown in Table 4, AdapNet++ outperforms all the baseline networks overall by a margin of 2.66% compared to the highest performing DeepLab v3 baseline which took 30,000 iterations more to reach this score. Unlike the performance in the Cityscapes and Synthia datasets where our previously proposed AdapNet architecture yields the second highest performance, AdapNet is outperformed by DeepLab v3 in the SUN RGB-D dataset. AdapNet++ on the other hand, outperforms the baselines in most categories by a large margin, while it is outperformed in 13 of the 37 classes by small margin. It can also be observed that the classes in which AdapNet++ get outperformed are the most infrequent classes. This can be alleviated by adding supplementary train-

ing images containing the low-frequency classes from other datasets or by employing class balancing techniques. However, our initial experiments employing techniques such as median frequency class balancing, inverse median frequency class balancing, normalized inverse frequency balancing, severely affected the performance of our model.

We report results on the ScanNet validation set in Table 5. AdapNet++ outperforms the state-of-the-art overall by a margin of 2.83%. The large improvement can be attributed to the proposed eASPP which efficiently captures long range context. Context aggregation plays an important role in such cluttered indoor datasets as different parts of an object are occluded from different viewpoints and across scenes. As objects such as the legs of a chair have thin structures, multi-scale learning contributes to recovering such structures. We see a similar trend in the performance as in the SUN RGB-D dataset, where our network outperforms the baselines in most of the object categories (16 of the 20 classes) significantly, while yielding a comparable performance for the other categories. The largest improvement of 5.70% is obtained for

**Table 4** Performance comparison of AdapNet++ with baseline models on the SUN RGB-D validation set (input image dim: 768 × 384) (Color table online)

Network	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Bshelf	Picture	Counter	Blinds	Desk
FCN-8s	68.57	79.94	37.27	52.85	58.42	42.79	43.69	27.58	43.49	28.46	42.21	27.08	28.34	11.38
SegNet	70.18	82.01	37.62	45.77	57.22	35.86	40.60	30.32	39.37	29.67	40.67	22.48	32.13	14.79
FastNet	68.39	79.73	34.98	44.93	55.26	36.68	38.57	26.88	38.28	26.02	39.72	20.55	22.47	11.71
ParseNet	70.83	81.85	40.25	55.64	61.12	45.06	45.86	30.41	41.73	<b>34.45</b>	44.11	28.63	32.59	12.78
DeconvNet	61.53	75.52	26.97	38.52	48.46	30.94	37.12	21.99	27.48	0.00	0.00	0.00	0.00	9.40
DeepLab v2	63.68	74.07	31.15	47.00	55.46	40.81	40.60	29.10	37.95	18.67	35.67	23.20	25.08	13.52
AdapNet	72.82	<b>86.61</b>	41.90	56.63	64.99	46.14	46.74	36.14	44.43	28.40	<b>46.53</b>	25.17	31.07	13.41
DeepLab v3	<b>74.99</b>	85.44	39.64	54.14	64.23	45.84	47.35	38.30	45.50	30.70	45.30	15.21	24.92	18.18
AdapNet++ (ours)	73.81	84.79	<b>47.45</b>	<b>64.31</b>	<b>65.76</b>	<b>52.15</b>	<b>49.97</b>	<b>41.66</b>	<b>46.99</b>	32.83	45.19	<b>32.71</b>	<b>34.72</b>	<b>21.16</b>

Network	Shelves	Curtain	Dresser	Pillow	Mirror	FMat	Clothes	Ceiling	Books	Fridge	Tv	Paper	Towel	ShwrC
FCN-8s	6.01	46.96	28.80	27.73	18.97	0.00	20.34	51.98	30.98	9.06	<b>34.23</b>	15.95	12.79	0.00
SegNet	5.15	39.69	27.46	26.05	15.64	0.00	16.01	53.89	26.21	12.95	24.23	14.57	10.88	0.00
FastNet	5.67	41.98	25.08	22.76	11.76	0.10	13.37	49.14	29.38	17.49	19.80	17.95	13.55	2.93
ParseNet	<b>8.58</b>	46.51	34.05	28.07	24.82	0.02	19.10	52.28	<b>34.36</b>	27.05	31.49	<b>21.65</b>	21.49	<b>6.46</b>
DeconvNet	0.00	29.59	0.00	0.00	8.32	0.00	0.00	46.40	0.00	0.00	0.00	0.00	0.00	0.00
DeepLab v2	2.55	42.91	24.56	29.81	<b>27.16</b>	0.00	15.85	43.47	25.39	35.55	23.21	16.53	22.00	0.34
AdapNet	3.98	47.90	28.58	<b>32.90</b>	19.25	0.00	17.30	52.60	30.63	20.32	16.56	18.21	13.69	0.00
DeepLab v3	66.37	48.84	28.20	32.87	31.23	0.00	16.98	60.09	31.74	37.86	33.80	17.84	20.51	0.00
AdapNet++ (ours)	5.03	<b>50.32</b>	<b>37.45</b>	32.35	25.89	<b>0.12</b>	<b>21.68</b>	<b>60.65</b>	32.41	<b>46.96</b>	18.81	20.79	<b>27.95</b>	1.32

Network	Box	Wboard	Person	NStand	Toilet	Sink	Lamp	Bathtub	Bag	mIoU (%)
FCN-8s	18.24	41.38	8.92	0.97	59.64	44.21	22.83	26.20	8.81	30.46
SegNet	13.50	38.02	3.83	0.76	60.98	45.97	22.82	31.94	8.97	29.14
FastNet	15.33	38.12	15.82	7.93	54.08	39.59	22.20	23.71	9.80	28.15
ParseNet	20.77	44.48	34.59	15.37	66.96	47.15	31.49	28.14	13.19	34.68
DeconvNet	4.43	32.06	0.00	0.00	47.35	32.85	0.00	0.00	0.00	15.64
DeepLab v2	13.56	22.32	<b>46.18</b>	11.45	55.94	43.33	29.52	32.73	7.63	30.06
AdapNet	15.72	43.02	17.48	9.44	61.11	49.43	31.32	21.56	9.38	32.47
DeepLab v3	19.58	44.10	31.36	18.15	68.76	52.01	35.36	<b>45.97</b>	10.37	35.74
AdapNet++ (ours)	<b>22.70</b>	<b>51.08</b>	33.77	<b>20.59</b>	<b>71.28</b>	<b>55.31</b>	<b>36.74</b>	38.60	<b>15.34</b>	<b>38.40</b>

Note that no left–right flips or multiscale testing is performed. Corresponding multimodal results are reported in Table 19. Bold numbers denote the highest performance among the compared methods.



**Table 5** Performance comparison of AdapNet++ with baseline models on the ScanNet validation set (input image dim: 768 × 384) (Color table online)

Network	Wall	Floor	Cabinet	Bed	Chair	Sofa	Table	Door	Window	Bshelf	Picture	Counter	Desk	Curtain
FCN-8s	70.00	74.70	39.31	66.23	51.96	48.21	45.19	42.61	42.15	<b>62.59</b>	<b>27.66</b>	24.69	33.25	<b>48.29</b>
SegNet	64.03	63.46	28.44	30.22	39.84	27.09	35.68	31.86	29.36	35.23	21.19	12.08	16.45	14.44
FastNet	66.31	71.35	34.22	56.32	47.86	44.87	39.34	33.20	34.32	50.11	23.27	24.48	28.87	42.56
ParseNet	<b>71.29</b>	76.42	39.92	71.11	52.72	51.07	48.74	43.06	45.25	58.81	31.33	27.84	38.46	54.81
DeconvNet	66.41	69.83	30.57	51.30	38.27	42.98	38.74	32.51	28.86	22.31	0.00	0.00	19.70	31.62
DeepLab v2	59.80	74.21	39.23	50.21	48.95	46.25	47.39	41.27	34.32	48.27	23.36	29.26	35.13	33.89
AdapNet	70.62	76.73	43.89	54.10	50.21	48.58	51.95	49.33	35.75	50.05	26.41	30.93	37.23	34.64
DeepLab v3	70.83	77.65	44.21	55.38	54.36	50.23	54.26	51.13	36.14	51.48	25.78	31.59	38.11	36.20
AdapNet+++ (ours)	71.21	<b>80.41</b>	<b>46.48</b>	<b>60.72</b>	<b>58.89</b>	<b>55.12</b>	<b>58.42</b>	<b>55.15</b>	<b>37.18</b>	52.15	26.73	<b>32.79</b>	<b>39.66</b>	37.12

Network	Fridge	SCurtain	Toilet	Sink	Bathtub	OtherF	mIoU (%)
FCN-8s	51.96	32.15	55.67	35.80	38.27	26.72	45.87
SegNet	0.09	24.34	30.70	32.50	3.47	9.73	27.51
FastNet	37.77	20.27	49.55	27.27	31.99	24.39	39.42
ParseNet	52.40	28.60	58.53	36.27	41.91	25.85	47.72
DeconvNet	1.31	0.12	0.00	0.00	0.00	16.66	24.56
DeepLab v2	43.69	42.57	62.26	39.97	55.40	22.36	43.89
AdapNet	43.27	45.83	68.22	43.45	60.37	24.04	47.28
DeepLab v3	51.76	51.46	76.22	51.24	64.36	29.32	50.09
AdapNet+++ (ours)	<b>54.54</b>	<b>54.88</b>	<b>81.92</b>	<b>54.91</b>	<b>68.65</b>	<b>31.46</b>	<b>52.92</b>

Note that no left–right flips or multiscale testing is performed. Corresponding multimodal results are reported in Table 20. Bold numbers denote the highest performance among the compared methods

**Table 6** Benchmarking results on the ScanNet test set with full resolution evaluation

Network	Multimodal	mIoU (%)
Enet	–	37.6
PSPNet	–	47.5
3DMV (2d proj)	✓	49.8
FuseNet	✓	52.1
AdapNet++ (ours)	–	<b>50.3</b>
SSMA (ours)	✓	<b>57.7</b>

Results were obtained from the ScanNet benchmark leaderboard  
 Bold numbers indicate the best performance among the compared methods

the *toilet* class, followed by an improvement of 5.34% for the *bed* class which appears as many different variations in the dataset. An interesting observation that can be made is that the highest parametrized network DeconvNet which has 252M parameters has the lowest performance in both SUN RGB-D and ScanNet datasets, while AdapNet++ which has about 1/9th of the parameters, outperforms it by more than twice the margin. However, this is only observed in the indoor datasets, while in the outdoor datasets DeconvNet performs comparable to the other networks. This is primarily due to the fact that indoor datasets have more number of small classes and the predictions of DeconvNet do not retain them.

Table 6 shows the results on the ScanNet test set. We compare against the top performing models on the leaderboard, namely, FuseNet (Hazirbas et al. 2016), 3DMV (2d proj) (Dai and Nießner 2018), PSPNet (Zhao et al. 2017), and Enet (Paszke et al. 2016). Note that 3DMV and FuseNet are multimodal fusion methods. Our proposed AdapNet++ model outperforms all the unimodal networks and achieves state-of-the-art performance for unimodal semantic segmentation on the ScanNet benchmark.

Finally, we also benchmark on the Freiburg Forest dataset as it contains several modalities and it is the largest dataset to provide labeled training data for unstructured forested environments. We show the results on the Freiburg Forest dataset in Table 7, where our proposed AdapNet++ outperforms the state-of-the-art by 0.82%. Note that this dataset contains large objects such trees and it does not contain thin structures or objects in multiple scales. Therefore, the improvement produced by AdapNet++ is mostly due to the proposed decoder which yields an improved resolution of segmentation along the object boundaries. The actual utility of this dataset is seen in the qualitative multimodal fusion results, where the fusion helps to improve the segmentation in the presence of disturbances such as glare on the optics and snow. Nevertheless, we see the highest improvement of 3.52% in the *obstacle* class, which is the hardest to segment in this dataset as it contains many different types of objects in one category and it has comparatively fewer examples in the dataset

Moreover, we also compare the number of parameters and the inference time with the baseline networks in Table 7. Our proposed AdapNet++ architecture performs inference in 72.77ms on an NVIDIA TITAN X which is substantially faster than the top performing architectures in all the benchmarks. Most of them consume more than twice the amount of time and the number of parameters making them unsuitable for real-world resource constrained applications. Our critical design choices enable AdapNet++ to consume only 10.98ms more than our previously proposed AdapNet, while exceeding its performance in each of the benchmarks by a large margin. This shows that AdapNet++ achieves the right performance vs. compactness trade-off which enables it to be employed in not only resource critical applications, but also in applications that demand efficiency and a fast inference time.

**Table 7** Performance comparison of AdapNet++ with baseline models on the Freiburg Forest validation set (input image dim: 768 × 384) (Color table online)

Network	Trail ■	Grass ■	Veg. ■	Sky ■	Obst. ■	mIoU (%)	gIoU (%)	FPR (%)	FNR (%)	Params. (M)	Time (ms)
FCN-8s	82.60	85.69	88.78	89.97	40.40	77.49	86.64	6.21	7.15	134.0	101.99
SegNet	82.12	84.99	88.64	89.90	27.23	74.58	86.24	6.77	6.98	29.4	79.13
ParseNet	85.67	87.33	89.63	89.17	43.08	78.97	87.65	6.34	6.01	20.5	286.54
FastNet	85.70	87.53	90.23	90.73	43.76	79.67	88.18	6.09	5.72	21.0	49.31
DeconvNet	86.37	87.17	89.63	92.20	34.80	78.04	89.06	6.53	6.78	252.0	168.54
DeepLab v2	88.75	88.87	90.29	91.65	49.55	81.82	89.98	5.94	5.79	43.7	128.90
AdapNet	88.99	88.99	91.18	92.89	48.80	82.17	90.67	4.89	5.42	24.4	61.81
DeepLab v3	88.62	88.84	90.55	91.75	51.61	82.28	90.08	5.21	5.82	58.16	82.83
AdapNet++ (ours)	<b>89.27</b>	<b>89.41</b>	<b>91.43</b>	<b>93.01</b>	<b>52.32</b>	<b>83.09</b>	<b>90.75</b>	<b>4.84</b>	<b>5.37</b>	<b>28.1</b>	<b>72.77</b>

Note that no left–right flips or multiscale testing is performed. Corresponding multimodal results are reported in Table 21  
 Bold numbers indicate the best performance among the compared methods

**Table 8** Comparison of network compression approaches on our AdapNet++ model trained on the Cityscapes dataset and evaluated on the validation set

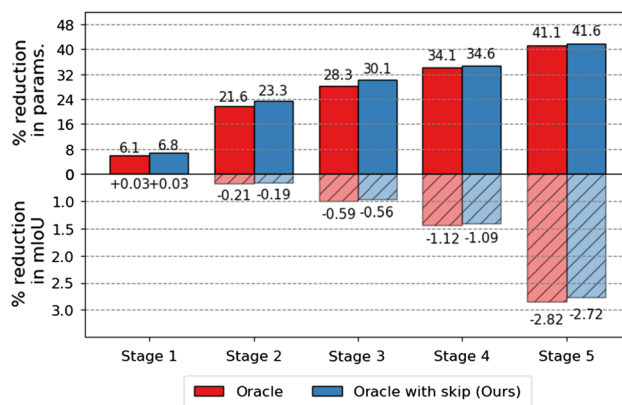
Technique	mIoU (%)	Param. (M)	FLOPS (B)	Reduction % of	
				Param.	FLOPS
Original	80.77	30.20	138.47	–	–
Baseline	80.67	28.57	136.05	–5.40	–1.75
Oracle	80.80	28.34	135.64	–6.15	–2.04
	80.56	23.67	125.33	–21.62	–9.49
	80.18	21.66	83.84	–28.28	–39.45
	79.65	19.91	81.72	–34.07	–40.98
	77.95	17.79	79.84	–41.09	–42.34
Oracle with skip (ours)	<b>80.80</b>	28.14	135.17	– <b>6.82</b>	– <b>2.38</b>
	<b>80.58</b>	23.16	124.14	– <b>23.31</b>	– <b>10.34</b>
	<b>80.21</b>	21.11	83.01	– <b>30.10</b>	– <b>40.05</b>
	<b>79.68</b>	19.75	81.53	– <b>34.60</b>	– <b>41.12</b>
	<b>78.05</b>	17.63	79.48	– <b>41.62</b>	– <b>42.60</b>

Bold numbers denote the highest performance among the compared methods

### 5.4 AdapNet++ Compression

In this section, we present empirical evaluations of our proposed pruning strategy that is invariant to shortcut connections in Table 8. We experiment with pruning entire convolutional filters which results in the removal of its corresponding feature map and the related kernels in the following layer. Most existing approaches only prune the first and the second convolution layer of each residual block, or in addition, equally prune the third convolution layer similar to the shortcut connection. However, this equal pruning strategy always leads to a significant drop in the accuracy of the model that is not recoverable (Li et al. 2016). Therefore, recent approaches have resorted to omitting pruning of these connections. Contrarily, our proposed technique is invariant to the presence of identity or projection shortcut connections, thereby making the pruning more effective and flexible. We employ a greedy pruning approach but rather than pruning layer by layer and fine-tuning the model after each step, we perform pruning of entire residual blocks at once and then perform the fine-tuning. As our network has a total of 75 convolutional and deconvolutional layers, pruning and fine-tuning each layer will be extremely cumbersome. Nevertheless, we expect a higher performance employing a fully greedy approach.

We compare our strategy with a baseline approach (Li et al. 2016) that uses the  $\ell^1$ -norm of the convolutional filters to compute their importance as well as the approach that we build upon that uses the Taylor expansion criteria (Molchanov et al. 2017) for the ranking as described in Sect. 3.5. We denote the approach of Molchanov et al. (2017) as Oracle in our results. In the first stage, we start by pruning only the *Res5* block of our model as it contains the most number of filters, therefore, a substantial amount of param-



**Fig. 15** Evaluation of network compression approaches shown as the percentage of reduction in the number of parameters with the corresponding decrease in the mIoU score for various baseline approaches versus our proposed technique. The results are shown for the AdapNet++ model trained on the Cityscapes dataset and evaluated on the validation set (Color figure online)

ters can be reduced without any loss in accuracy. As shown in Table 8, our approach enables a reduction of 6.82% of the parameters and 3.3B FLOPS with a slight increase in the mIoU metric. Similar to our approach the original Oracle approach does not cause a drop in the mIoU metric but achieves a lower reduction in parameters. Whereas, the baseline approach achieves a smaller reduction in the parameters and simultaneously causes a drop in the mIoU score.

Our aim for pruning in the first stage was to compress the model without causing a drop in the segmentation performance, while in the following stages, we aggressively prune the model to achieve the best parameter to performance ratio. Results from this experiment are shown as the percentage in reduction of parameters in comparison to the change in mIoU in Fig. 15. In the second stage, we prune the convolutional

**Table 9** Effect of the various contributions proposed in the AdapNet++ architecture

Model	Encoder	MS Residual	Decoder	Skip	ASPP dil 3, 6, 12	Aux Loss 2×	He Init	mIoU (%)
M1	ResNet-50	–	$c_n^1 \uparrow_n^{16}$	–	–	–	–	75.22
M2	ResNet-50	✓	$c_n^1 \uparrow_n^{16}$	–	–	–	–	76.92
M3	ResNet-50	✓	$c_n^1 \uparrow_{2n}^2 \oplus \uparrow_n^8$	3d	–	–	–	77.78
M4	PA ResNet-50	✓	$c_n^1 \uparrow_{2n}^2 \oplus \uparrow_n^8$	3d	–	–	–	78.44
M5	PA ResNet-50	✓	$c_n^1 \uparrow_{2n}^2 \oplus \uparrow_n^8$	3d	✓	–	–	78.93
M6	PA ResNet-50	✓	$c_n^1 \uparrow_{2n}^2 \oplus \uparrow_{2n}^2 \oplus \uparrow_n^8$	3d,2c	✓	–	–	79.19
M7	PA ResNet-50	✓	$\uparrow^2 \parallel c^3 c^3 \uparrow^2 \parallel c^3 c^3 c_n^1 \uparrow_n^8$	3d,2c	✓	–	–	79.82
M8	PA ResNet-50	✓	$\uparrow^2 \parallel c^3 c^3 \uparrow^2 \parallel c^3 c^3 c_n^1 \uparrow_n^8$	3d,2c	✓	✓	–	80.34
M9	PA ResNet-50	✓	$\uparrow^2 \parallel c^3 c^3 \uparrow^2 \parallel c^3 c^3 c_n^1 \uparrow_n^8$	3d,2c	✓	✓	✓	<b>80.67</b>

The performance is shown for the model trained on the Cityscapes dataset and evaluated on the validation set.  $\uparrow_f^k$  refers to a deconvolution layer and  $c_f^k$  refers to a convolution layer with  $f$  number of filters,  $k \times k$  kernel size and  $n$  is the number of classes. PA ResNet-50 refers to the full preactivation ResNet-50 architecture. The weights for the two auxiliary losses were set to  $\mathcal{L}_1 = 0.5$  and  $\mathcal{L}_2 = 0.6$  for this experiment. Bold numbers denote the highest performance among the compared methods.

feature maps of *Res2*, *Res3*, *Res4* and *Res5* layers. Using our proposed method, we achieve a reduction of 23.31% of parameters with minor drop of 0.19% in the mIoU score. Whereas, the Oracle approach yields a lower reduction in parameters as well as a larger drop in performance. A similar trend can also be seen for the other pruning stages where our proposed approach yields a higher reduction in parameters and FLOPS with a minor reduction in the mIoU score. This shows that pruning convolutional feature maps with regularity leads to a better compression ratio than selectively pruning layers at different stages of the network. In the third stage, we perform pruning of the deconvolutional feature maps, while in the fourth and fifth stages we further prune all the layers of the network by varying the threshold for the rankings. In the final stage we obtain a reduction of 41.62% of the parameters and 42.60% of FLOPS with a drop of 2.72% in the mIoU score. Considering the compression that can be achieved, this minor drop in the mIoU score is acceptable to enable efficient deployment in resource constrained applications.

## 5.5 AdapNet++ Ablation Studies

In order to evaluate the various components of our AdapNet++ architecture, we performed several experiments in different settings. In this section, we study the improvement obtained due to the proposed encoder with the multiscale residual units, a detailed analysis of the proposed eASPP, comparisons with different base encoder network topologies, the improvement that can be obtained by using higher resolution images as input and using multiscale testing. For each of these components, we also study the effect of different parameter configurations. All the ablation studies presented in this section were performed on models trained on the Cityscapes dataset.

### 5.5.1 Detailed Study on the AdapNet++ Architecture

We first study the major contributions made to the encoder as well as the decoder in our proposed AdapNet++ architecture. Table 9 shows results from this experiment and subsequent improvement due to each of the configurations. The simple base model M1 consisting of the standard ResNet-50 for the encoder and a single deconvolution layer for upsampling achieves a mIoU of 75.22%. The model M2 that incorporates our multiscale residual units achieves an improvement of 1.7% without any increase in the memory consumption. Whereas, the multigrid approach from DeepLab v3 (Chen et al. 2017) in the same configuration achieves only 0.38% of improvement in the mIoU score. This shows the novelty in employing our multiscale residual units for efficiently learning multiscale features throughout the network. In the M3 model, we study the effect of incorporating skip connections for refinement. Skip connections that were initially introduced in the FCN architecture are still widely used for improving the resolution of the segmentation by incorporating low or mid-level features from the encoder into the decoder while upsampling. The ResNet-50 architecture contains the most discriminative features in the middle of the network. In our M3 model, we first upsample the encoder output by a factor two, followed by fusing the features from *Res3d* block of the encoder to refinement and subsequently using another deconvolutional layer to upsample back to input resolution. This model achieves a further improvement of 0.86%.

In the M4 model, we replace the standard residual units with the full pre-activation residual units which yields an improvement of 0.66%. As mentioned in the work by He et al. (2016), the results corroborate that pre-activation residual units yields a lower error than standard residual units due



**Table 10** Evaluation of various atrous spatial pyramid pooling configurations

Model	ASPP conv.	Decoder conv.	mIoU (%)	Param (M)	FLOPS (B)
M91	$[c_{256}^3]$	$[c_{256}^1]$	80.06	41.3	115.99
M92	$[c_{256}^3]$	$[c_{256}^3]$	80.27	42.5	142.42
M93	$[c_{256}^3]$	$[c_{256}^3] \times 2$	80.67	43.7	169.62
M94	$[c_{64}^1 c_{64}^3 c_{256}^1]$	$[c_{256}^3] \times 2$	80.42	30.1	138.21
M95	$[c_{64}^1 c_{64}^3 c_{64}^3 c_{256}^1]$	$[c_{256}^3] \times 2$	<b>80.77</b>	<b>30.2</b>	<b>138.47</b>

The performance is shown for the model trained on the Cityscapes dataset and evaluated on the validation set.  $\uparrow_f^k$  refers to a deconvolution layer and  $c_f^k$  refers to a convolution layer with  $f$  number of filters and  $k \times k$  kernel size. The weights for the two auxiliary losses were set to  $\mathcal{L}_1 = 0.5$  and  $\mathcal{L}_2 = 0.6$  for this experiment

Bold numbers indicate the best performance among the compared methods

to the ease of training and improved generalization capability. Aggregating multiscale context using ASPP has become standard practice in most classification and segmentation networks. In the M5 model, we add the ASPP module to the end of the encoder segment. This model demonstrates an improved mIoU of 78.93% due to the ability of the ASPP to capture long range context. In the subsequent M6 model, we study if adding another skip refinement connection from the encoder yields a better performance. This was challenging as most combinations along with the *Res3d* skip connection did not demonstrate any improvement. However, adding a skip connection from *Res2c* showed a slight improvement.

In all the models upto this stage, we fused the mid-level encoder features into the decoder using element-wise addition. In order to make the decoder stronger, we experiment with improving the learned decoder representations with additional convolutions after concatenation of the mid-level features. Specifically, the M7 model has three upsampling stages, the first two stages consist of a deconvolution layer that upsamples by a factor of two, followed by concatenation of the mid-level features and two following  $3 \times 3$  convolutions that learn highly discriminative fused features. This model shows an improvement of 0.63% which is primarily due to the improved segmentation along the object boundaries as demonstrated in the qualitative results in Sect. 5.11. Our M7 model contains a total of 75 convolutional and deconvolutional layers, making the optimization challenging. In order to accelerate the training and to further improve the segmentation along object boundaries, we propose a multiresolution supervision scheme in which we add a weighted auxiliary loss to each of the first two upsampling stages. This model denoted as M8 achieves an improved mIoU of 80.34%. In comparison to aforementioned scheme, we also experimented with adding a weighted auxiliary loss at the end of the encoder of the M7 model, however it did not improve the performance, although it accelerated the training. Finally we also experimented with initializing the layers with the He initialization (He et al. 2015b) scheme (also known as MSRA) in the M9 model which further boosts the mIoU to 80.67%.

The following section further builds upon the M9 model to yield the topology of our proposed AdapNet++ architecture.

### 5.5.2 Detailed Study on the eASPP

In this section, we quantitatively and qualitatively evaluate the performance of our proposed eASPP configuration and the new decoder topology. We perform all the experiments in this section using the best performing M9 model described in Sect. 5.5.1 and report the results on the Cityscapes validation set in Table 10. In the first configuration of the M91 model, we employ a single  $3 \times 3$  atrous convolution in the ASPP, similar the configuration proposed in DeepLab v3 (Chen et al. 2017) and use a single  $1 \times 1$  convolution in the place of the two  $3 \times 3$  convolutions in the decoder of the M9 model. This model achieves an mIoU score of 80.06% with 41.3M parameters and consumes 115.99B FLOPS. In order to better fuse the concatenated mid-level features with the decoder and to improve its discriminability, we replace the  $1 \times 1$  convolution layer with a  $3 \times 3$  convolution in the M92 model and with two  $3 \times 3$  convolutions in the M93 model. Both these models demonstrate an increase in performance corroborating that a simple  $1 \times 1$  convolution is insufficient for object boundary refinement using fusion of mid-level encoder features.

In an effort to reduce the number of parameters, we employ a bottleneck architecture in the ASPP of the M94 model by replacing the  $3 \times 3$  atrous convolution with a structure consisting of a  $1 \times 1$  convolution with half the number of filters, followed by a  $3 \times 3$  atrous convolution with half the number of filters and another  $1 \times 1$  convolution with the original amount of filters. This model achieves an mIoU score of 80.42% which accounts to a reduction of 0.25% in comparison to the M93 model, however, it reduces computational requirement by 13.6M parameters and 31.41B FLOPS which makes the model very efficient. Nevertheless, this drop in performance is not ideal. Therefore, in order to compensate for this drop, we leverage the idea of cascading atrous convolutions that enables an increase in the size of the effective receptive field and the density of the pixel sampling. Specifically, in the M95

**Table 11** Performance comparison of our proposed eASPP with various other ASPP configurations

ASPP topology	mIoU (%)	Param (M)	FLOPS (B)
ASPP v2	80.25	18.87	50.96
ASPP v3	80.67	15.53	34.58
ASPP v3 with separable conv.	80.27	3.00	5.56
DenseASPP	80.62	4.23	9.74
eASPP (ours)	<b>80.77</b>	<b>2.04</b>	<b>3.62</b>

The results are reported for the models trained on the Cityscapes dataset and evaluated on the validation set

Bold numbers indicate the best performance among the compared methods

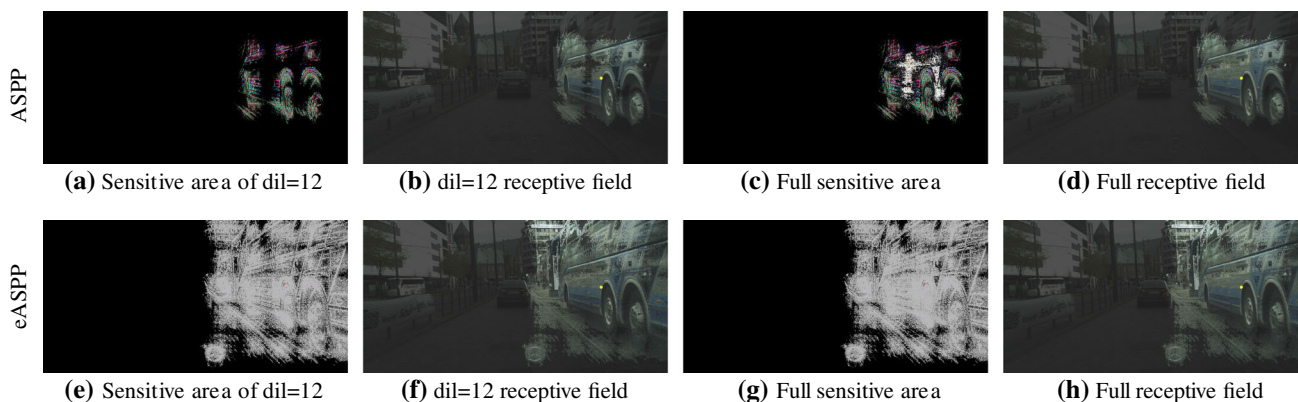
model, we add a cascaded  $3 \times 3$  atrous convolution in place of the single  $3 \times 3$  atrous convolution in the M94 model. This model achieves a mIoU score of 80.77% which is an increase of 0.35% in the mIoU with only a minor increase of 0.1M parameters in comparison to our M94 model. The originally proposed ASPP module consumes 15.53M parameters and 34.58B FLOPS, where the cascaded bottleneck structure in the M95 model only consumes 2.04M parameters and 3.62B FLOPS which is over 10 times more computationally efficient. We denote this cascaded bottleneck structure as eASPP.

Furthermore, we present detailed experimental comparisons of our proposed eASPP with other ASPP configurations in Table 11. Specifically, we compare against the initial ASPP configuration proposed in DeepLab v2 (Chen et al. 2016) which we denote as ASPP v2, the improved ASPP configuration that also incorporates image-level features as proposed in DeepLab v3 (Chen et al. 2017) which we denote as ASPP v3, the ASPP configuration with separable convolutions and the more recently proposed DenseASPP (Yang et al. 2018) configuration. In order to have a fair comparison, we use the same AdapNet++ architecture with the different ASPP configurations for this experiment and present the results on the Cityscapes validation set. The four parallel atrous convolution layers in the ASPP v2 configuration of DeepLab v2 have the number of feature channels equal to the number of object classes in the dataset, while the ASPP v3 configuration of DeepLab v3 has the number of feature channels equal to 256 in the three parallel atrous convolution layers.

The ASPP v2 model with the convolution feature channels set to the number of object classes achieves a mIoU of 79.22% and increasing the number of convolution feature channels to 256 yields a mIoU of 80.25%. By incorporating image-level features using a global pooling layer and removing the fourth parallel atrous convolution in ASPP v2, the ASPP v3 model achieves an improved performance of 80.67% with a minor decrease in the parameters and FLOPs. Recently, separable convolutions are being employed in place

of the standard convolution layer as an efficient alternative to reduce the model size. Employing atrous separable convolutions in the ASPP v3 configuration significantly reduces the number of parameters and FLOPs consumed by the model to 3M and 5.56B respectively. However, this also reduces the mIoU of the model to 80.27% which is comparable to the ASPP v2 configuration with 256 convolutional filters. The model with the DenseASPP configuration achieves a mIoU of 80.62% which is still lower than the ASPP v3 configuration but it reduces the number of parameters and FLOPs to 4.23M and 9.74B respectively. It should be noted that in the work of Yang et al. (2018), DenseASPP was only compared to ASPP v2 with the number of convolutional feature channels equal to the number of object classes (mIoU of 79.22%). In comparison to the aforementioned ASPP topologies, our proposed eASPP achieves the highest mIoU score of 80.77% with the lowest consumption of parameters and FLOPs. This accounts to a reduction of 86.86% of the number of parameters and 89.53% of FLOPs with a increase in the mIoU compared to the previously best performing ASPP v3 topology.

In order to illustrate the phenomenon caused by cascading atrous convolutions in our proposed eASPP, we visualize the empirical receptive field using the approach proposed by Zhou et al. (2014). First, for each feature vector representing an image patch, we use a  $8 \times 8$  mean image to occlude the patch at different locations using a sliding window. We then record the change in the activation by measuring the Euclidean distances as a heat map which indicates which regions are sensitive to the feature vector. Although the size of the empirical receptive fields is smaller than theoretical receptive fields, they are better localized and more representative of the information they capture (Zhou et al. 2014). In Fig. 16, we show visualizations of the empirical receptive field size of the convolution layer of the ASPP that has one  $3 \times 3$  atrous convolution in each branch in comparison to our M95 model that has cascaded  $3 \times 3$  atrous convolutions. Figure 16b, c show the receptive field at the annotated yellow dot for the atrous convolution with the largest dilation rate in ASPP and in our eASPP correspondingly. It can be seen that our eASPP has a much larger receptive field that enables capturing large contexts. Moreover, it can be seen that the pixels are sampled much denser in our eASPP in comparison to the ASPP. In Fig. 16d, h, we show the aggregated receptive fields of the entire module in which it can be observed that our eASPP has much lesser isolated points of focus and a cleaner sensitive area than the ASPP. We evaluated the generalization of our proposed eASPP by incorporating the module into our AdapNet++ architecture and benchmarking its performance in comparison to DeepLab which incorporates the ASPP. The results presented in Sect. 5.3 demonstrate that our eASPP effectively generalizes to a wide range of datasets containing diverse environments.



**Fig. 16** Comparison of the receptive field of ASPP and our proposed eASPP. The receptive field is visualized for the annotated yellow dot. Our proposed eASPP has larger receptive field size and denser pixel sampling in comparison to the ASPP (Color figure online)

**Table 12** Effect on varying the number of filters in the skip refinement connections in the M95 model

Skip channels	12	24	36	48	60
mIoU (%)	80.50	<b>80.77</b>	80.67	80.59	80.56

The performance is shown for the model trained on the Cityscapes dataset and evaluated on the validation set

Bold numbers denote the highest performance among the compared methods

### 5.5.3 Improving Granularity of Segmentation

In our AdapNet++ architecture, we propose two strategies to improve the segmentation along object boundaries in addition to the new decoder architecture. The first being the two skip refinement stages that fuse mid-level encoder features from *Res3d* and *Res2c* into the decoder for object boundary refinement. However, as the low and mid-level features have a large number of filters (512 in *Res3d* and 256 in *Res2c*) in comparison to the decoder filters that only have 256 feature channels, they will outweigh the high level features and decrease the performance. Therefore, we employ a  $1 \times 1$  convolution to reduce the number of feature channels in the low and mid-level representations before fusing them into the decoder. In Table 12, we show results varying the number of feature channels in the  $1 \times 1$  skip refinement convolutions in the M95 model from Sect. 5.5.2. We obtain the best results by reducing the number of mid-level encoder feature channels to 24 using the  $1 \times 1$  convolution layer.

The second strategy that we employ for improving the segmentation along object boundaries is using our proposed multiresolution supervision scheme. As described in Sect. 3.4, we employ auxiliary loss branches after each of the first two upsampling stages in the decoder to improve the resolution of the segmentation and to accelerate training. Weighing the two auxiliary losses is critical to balance the gradient flow through all the previous layers of the network. We experiment with different loss weightings and report results for the same M95 model in Table 13. The network achieves the highest performance for auxiliary loss weightings  $\lambda_1 = 0.6$  and  $\lambda_2 = 0.5$  for  $\mathcal{L}_{aux1}$  and  $\mathcal{L}_{aux2}$  respectively.

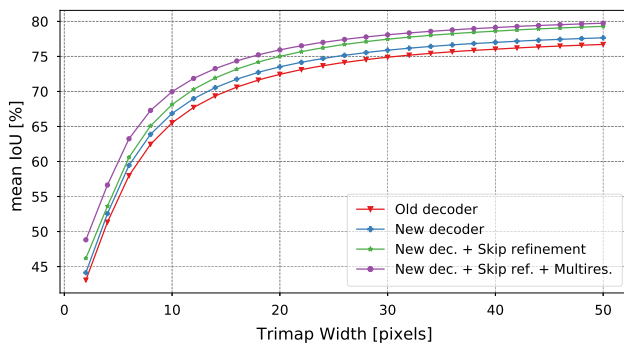
In order to quantify the improvement specifically along the object boundaries, we evaluate the performance of our architecture using the trimap experiment (Kohli and Torr 2009). The mIoU score for the pixels that are within the trimap band of the void class labels (255) are computed by applying the morphological dilation on the void labels. Results from this experiment shown in Fig. 17 demonstrates that our new decoder improves the performance along object boundaries compared to the decoder in AdapNet (Valada et al. 2017), while the M7 model with the new decoder and the skip refinement further improves the performance. Finally, the M8 model consisting of our new decoder with the skip refinement stages and our multiresolution supervision scheme for training significantly improves the segmentation along the boundaries which is more evident when the trimap band is narrow.

**Table 13** Effect on varying the weighting factor of the auxiliary losses in the M95 model

Aux 1 weight	0.3	0.4	0.4	0.5	0.5	0.5	0.6	0.6	0.6	0.7	0.7	1.0
Aux 2 weight	0.2	0.2	0.3	0.4	0.6	0.7	0.4	0.5	0.7	0.5	0.6	1.0
mIoU (%)	80.51	80.55	80.68	80.60	80.55	80.49	80.53	<b>80.77</b>	80.69	80.55	80.71	80.37

The performance is shown for the model trained on the Cityscapes dataset and evaluated on the validation set

Bold number denote the highest performance among the compared methods



**Fig. 17** Influence of the proposed decoder, skip refinement and multiresolution supervision on the segmentation along object boundaries using the trimap experiment. The plot shows the mIoU score as a function of the trimap band width along the object boundaries. The results are shown on the Cityscapes dataset and evaluated on the validation set (Color figure online)

**Table 14** Effect of various encoder topologies in the M95 model

Encoder	ResNet	PA ResNet	ResNeXt	SEnet	Xception
mIoU (%)	79.32	<b>80.77</b>	80.30	78.31	78.70
Param (M)	30.2	30.2	29.7	32.7	27.5
FLOPS (B)	135.28	138.47	145.81	145.34	137.06

The performance is shown for the model trained on the Cityscapes dataset and evaluated on the validation set

Bold number denote the highest performance among the compared methods

#### 5.5.4 Encoder Topology

In recent years, several efficient network architectures have been proposed for image classification that are computationally inexpensive and have fast inference times. In order to study the trade-off between accuracy and computational requirement, we performed experiments using five widely employed architectures as the encoder backbone. Specifically, we evaluate the performance using ResNet-50 (He et al. 2015a), full pre-activation ResNet-50 (He et al. 2016), ResNeXt (Xie et al. 2017), SEnet (Hu et al. 2017) and Xception (Chollet 2016) architectures for the encoder topology and augmented them with our proposed modules similar to the M95 model described in Sect. 5.5.2.

Results from this experiment are shown in Table 14. Note that in the comparisons presented in this section, no model compression has been performed. It can be seen that the full pre-activation ResNet-50 model achieves the highest mIoU score, closely followed by the ResNeXt model. However the ResNeXt model has an additional 7.34M parameters with a slightly lesser number of FLOPS. While, the standard ResNet-50 architecture has 3.19M parameters lesser than the full pre-activation ResNet-50 model, it achieves a lower mIoU score of 79.32%. Therefore, we choose

**Table 15** Performance of the strong decoder that we introduce in our AdapNet++ architecture in comparison with other progressive upsampling decoder topologies

Decoder topology	mIoU (%)	Param (M)	FLOPs (B)
Bilinear upsampling	74.83	—	—
LRR	79.38	2.01	75.88
RefineNet	80.44	14.29	265.33
AdapNet++ (ours)	<b>80.77</b>	4.59	69.84

All the models employ the same AdapNet++ encoder with only the decoder replaced with alternative topologies and the results are reported on the Cityscapes validation set

Bold number denote the highest performance among the compared methods

the full-preactivation ResNet-50 architecture as the encoder backbone in our proposed AdapNet++ architecture.

#### 5.5.5 Decoder Topology

In this section, we compare the performance and computational efficiency of the new strong decoder that we introduce in our proposed AdapNet++ architecture with other existing progressive upsampling decoders. For a fair comparison, we employ the same AdapNet++ encoder in all the models in this experiment and only replace the decoder with the topologies proposed in LRR (Ghiasi and Fowlkes 2016) and RefineNet (Lin et al. 2017). All these decoder topologies that we compare with utilize similar stage-wise upsampling with deconvolution layers and refinement with skip connections from higher resolution encoder feature maps. As a reference, we also compare with a model that employs the AdapNet++ encoder and direct bilinear upsampling for the decoder. Therefore, this reference model does not consume any parameters and FLOPs for the decoder section.

Table 15 shows the results from this experiment in which we see that the reference model with direct bilinear upsampling achieves a mIoU score of 74.83%. The LRR decoder model outperforms the reference model and the RefineNet decoder model further outperforms the LRR decoder model. However, the RefineNet decoder consumes a significant amount of parameters and FLOPs compared to the LRR decoder. Nevertheless, our proposed decoder in AdapNet++ achieves the highest mIoU score of 80.77%, thereby outperforming both the other competing progressive upsampling decoders while consuming the lowest amount of FLOPs and still maintaining a good parameter efficiency.

#### 5.5.6 Image Resolution and Testing Strategies

We further performed experiments using input images with larger resolutions as well as with left–right flipped inputs and multiscale inputs while testing. In all our benchmark-



**Table 16** Effect on using a higher resolution input image and employing left–right flip as well as multiscale inputs during testing

Image size (pixels)	Flip	MS	mIoU (%)	Acc. (%)	AP (%)	Time (ms)
768 × 384	–	–	80.77	96.04	90.97	72.77
768 × 384	✓	–	81.35	96.18	90.76	148.93
768 × 384	✓	✓	82.25	96.36	91.86	1775.96
896 × 448	–	–	81.69	96.06	89.96	88.89
896 × 448	✓	–	82.28	96.21	90.53	183.57
896 × 448	✓	✓	83.19	96.48	91.52	2342.31
1024 × 512	–	–	82.47	96.13	90.63	105.94
1024 × 512	✓	–	83.07	96.28	91.17	219.28
1024 × 512	✓	✓	84.27	96.66	92.36	3061.11
2048 × 1024	–	–	83.10	96.25	90.87	494.97
2048 × 1024	✓	–	83.58	96.37	91.14	1022.12
2048 × 1024	✓	✓	<b>84.54</b>	<b>96.74</b>	<b>92.48</b>	12188.57

The performance is shown for the model trained on the Cityscapes dataset and evaluated on the validation set. Bold number denote the highest performance among the compared methods

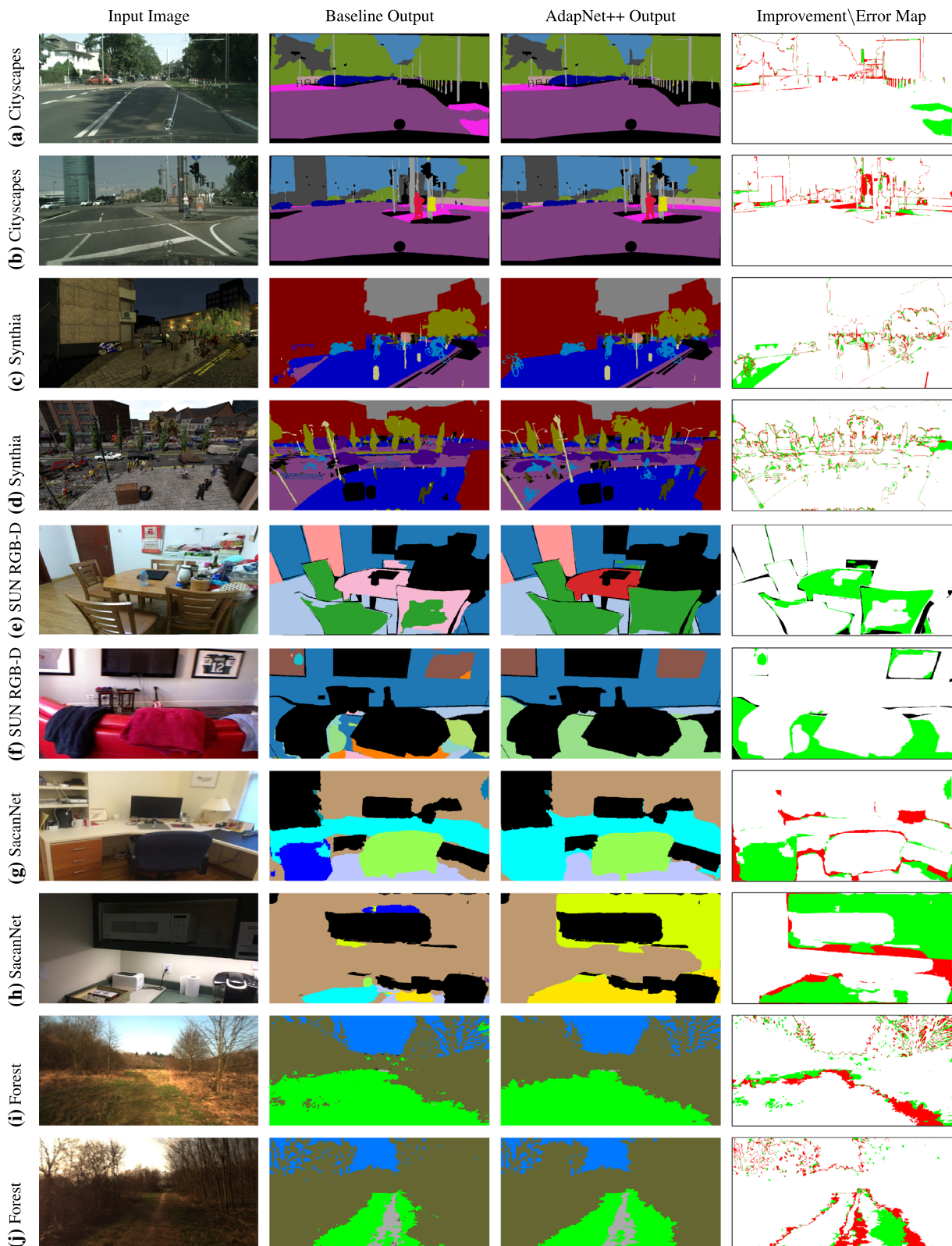
ing experiments, we use an input image with a resolution of  $768 \times 384$  pixels in order enable training of the multimodal fusion model that has two encoder streams on a single GPU. State-of-the-art semantic segmentation architectures use multiple crops from the full resolution of the image as input. For example for the Cityscapes dataset, eight crops of  $720 \times 720$  pixels from each full resolution image of  $2048 \times 1024$  pixels are often used. This yields a down-sampled output with a larger resolution at the end of the encoder, thereby leading to a lesser loss of information due to downsampling and more boundary delineation. Employing a larger resolution image as input also enables better segmentation of small objects that are at far away distances, especially in urban driving datasets such as Cityscapes. However, the caveat being that it requires multi-GPU training with synchronized batch normalization in order to utilize a large enough mini-batch size, which makes the training more cumbersome. Moreover, using large crops of the full resolution image significantly increases the inference time of the model as the inference time for one image is the sum of the inference time consumed for each of the crops.

Nevertheless, we present experimental results with input images of resolution  $896 \times 448$  pixels,  $1024 \times 512$  pixels, and eight crops of  $720 \times 720$  pixels from the full resolution of  $2048 \times 1024$  pixels, in addition to the resolution of  $768 \times 384$  pixels that we use. In addition to the varying input image resolutions, we also test with left–right flips and multiscale inputs. However, although this increases the mIoU score it substantially increases the computation complexity and runtime, therefore rendering it not useful in real-world applications. A summary of the results from this experiment are shown in Table 16. It can be seen that with each higher resolution image, the model yields an increased mIoU score and simultaneously consumes a larger inference time. Similarly, left–right flips and multiscale inputs also yield an improve-

ment in the mIoU score. For the input image resolution of  $768 \times 384$  pixels that we employ in the benchmarking experiments, left–right flips yields an increase of 0.58% in the mIoU, while multiscale inputs in addition, yields a further improvement of 0.9%. The corresponding pixel accuracy and average precision also shows an improvement. The model trained with eight crops of  $720 \times 720$  pixels from each full resolution image of  $2048 \times 1024$  pixels demonstrates an improvement of 2.33% in the mIoU score in comparison to the model with a lower resolution image that we use for benchmarking. Furthermore, using left–right flips and multiscale inputs yields an overall improvement of 3.77% in the mIoU and additional improvements in the other metrics in comparison to the benchmarking model. However, it can be seen that the inference time full resolution model is 494.98ms and multiscale testing with left–right flips further increases it to 12188.57ms, while the inference time of the model that uses a image resolution of  $768 \times 384$  pixels is only 72.77ms demonstrating that using full resolution of the image and multiscale testing with left–right flips for real-world robotics applications is impractical.

## 5.6 Qualitative Results of Unimodal Segmentation

In this section, we qualitatively evaluate the semantic segmentation performance of our AdapNet++ architecture in comparison to the best performing state-of-the-art model for each dataset according to the quantitative results presented in Sect. 5.3. We utilize this best performing model as a baseline for the qualitative comparisons presented in this section. Figure 18 shows two examples for each dataset that we benchmark on. The colors for the segmented labels shown correspond to the colors and the object categories mentioned in the benchmarking tables shown in Sect. 5.3. Figure 18a, b show examples from the Cityscapes dataset in which the



**Fig. 18** Qualitative segmentation results of our unimodal AdapNet++ architecture in comparison to the best performing state-of-the-art model on different datasets. In addition to the segmentation output, we also show the improvement/error map which denotes the misclassified pixels

in red and the pixels that are misclassified by the best performing state-of-the-art model but correctly predicted by AdapNet++ in green. The legend for the segmented labels correspond to the colors shown in the benchmarking tables in Sect. 5.3 (Color figure online)

improvement over the baseline output (AdapNet) can be seen in the better differentiation between inconspicuous classes such as *sidewalk* and *road* as well as *pole* and *sign*. This can be primarily attributed to the eASPP which has a large receptive field and thus captures larger object context which helps to discern the differences between the inconspicuous classes. The improvement due to better boundary segmentation of thin object classes such as *poles* can be seen in the images.

Figure 18c, d show examples from the Synthia dataset, where objects such as *bicycles*, *cars* and *people* are better segmented. The baseline output (AdapNet) shows several missing *cars*, *people* and *bicycles*, whereas the AdapNet++ output accurately captures these objects. Moreover, it can also be seen that the pole-like structures and trees are often discontinuous in the baseline output, while they are more well defined in the AdapNet++ output. In Fig. 18d, an interesting observation is made where an entire fence is segmented in the baseline output but is absent in the scene. This is due to the fact that the intersection of the sidewalk and the road gives an appearance of a fence which is then misclassified. In the same image, it can also be observed that a small building-like structure on the right is not captured, whereas our AdapNet++ model accurately segments the structure.

Figure 18e, f show examples from the indoor SUN RGB-D dataset. Examples from this dataset show significant misclassification due to inconspicuous objects. Often scenes in indoor datasets have large objects that require the network to have very large receptive fields to be able to accurately distinguish between them. Figure 18e shows a scene in which parts of the *chair* and the *table* are incorrectly classified as a *desk* in the output of the baseline model (DeepLab v3). These two classes have very similar structure and appearance which makes distinguishing between them extremely challenging. In Fig. 18f, we can see parts of the *sofa* incorrectly classified in the baseline model output, whereas the entire object is accurately predicted in the AdapNet++ output. In the baseline output, misclassification can also be seen for the *picture* on the wall which is precisely segmented in the AdapNet++ output.

In Fig. 18g, h, we show examples from the indoor ScanNet dataset. Figure 18g shows misclassification in the output of the baseline model (DeepLab v3) in the boundary where the wall meets the floor and for parts of the *desk* that is misclassified as other furniture. Figure 18h shows a significant improvement in the segmentation of AdapNet++ in comparison to the baseline model. The *cabinet* and *counter* are entirely misclassified as a *desk* and *other furniture* correspondingly in the output of the baseline model, whereas they are accurately predicted by our AdapNet++ mode.

Finally, Fig. 18i, j show examples from the unstructured Freiburg Forest dataset where the improvement can largely be seen in discerning the object boundaries of classes such as

*grass* and *vegetation*, as well as *trail* and *grass*. By observing these images, we can see that even for us humans it is difficult to estimate the boundaries between these classes. Our AdapNet++ architecture predicts the boundaries comparatively better than the baseline model (DeepLab v3). The improvement in the segmentation can also be seen in the finer segmentation of the *vegetation* and the *trail* path in the AdapNet++ output.

## 5.7 Multimodal Fusion Benchmarking

In this section, we present comprehensive results on the performance of our proposed multimodal SSMA fusion architecture in comparison to state-of-the-art multimodal fusion methods, namely, LFC (Valada et al. 2016b), FuseNet (Hazirbas et al. 2016) and CMoDE (Valada et al. 2017). We employ the same AdapNet++ network backbone for all the fusion models including the competing methods. Therefore, we use the official implementation from the authors as a reference and append the fusion mechanism to our backbone. We also compare with baseline fusion approaches with the AdapNet++ topology as the backbone such as Late Fusion: a  $1 \times 1$  convolution layer appended after individual modality-specific networks and the outputs are merged by adding the feature maps before the softmax, Stacking: channel-wise concatenation of modalities before input to the network, Average: averaging prediction probabilities of individual modality-specific networks followed by *argmax*, and Maximum: maximum of the prediction probabilities of individual modality-specific networks followed by *argmax*. Additionally, we also compare against the performance of the unimodal AdapNet++ architecture for each of the modalities in the dataset for reference. We denote our proposed multimodal model as SSMA and the model with left–right flips as well as multiscale testing as SSMA\_msf in our experiments.

In Table 17, we show the results on the Cityscapes validation set considering visual images (RGB), depth and the HHA encoding of the depth as modalities for the fusion. As hypothesised, the visual RGB images perform the best among the other modalities achieving a mIoU of 80.80%. This is especially observed in outdoor scene understanding datasets containing stereo depth images that quickly degrade the information contained, with increasing distance from the camera. Among the baseline fusion approaches, Stacking achieves the highest performance for both RGB-D and RGB-HHA fusion, however, their performance is still lower than the unimodal visual RGB segmentation. This can be attributed to the fact that the baseline approaches are not able to exploit the complementary features from the modalities due to the naive fusion. CMoDE fusion with RGB-HHA achieves the highest performance among state-of-the-art approaches, surpassing the performance of unimodal segmentation. While, our proposed SSMA model for RGB-HHA

**Table 17** Comparison of multimodal fusion approaches on the Cityscapes validation set (input image dim:  $768 \times 384$ )

Network	Approach	mIoU (%)	Acc. (%)	AP (%)	
RGB	Unimodal	80.80	96.04	90.97	
	Depth	66.36	91.21	80.23	
	HHA	67.66	91.66	81.81	
	RGB-D	Average	78.84	95.58	90.49
		Maximum	78.81	95.58	90.37
		Stacking	80.21	95.96	90.05
		Late Fusion	78.75	95.57	90.48
		LFC	81.04	96.11	91.10
		CMoDE	81.33	96.12	90.29
	RGB-HHA	FuseNet	81.54	96.23	90.24
SSMA (ours)		82.29	96.36	90.77	
SSMA_msf (ours)		<b>83.44</b>	<b>96.59</b>	<b>92.21</b>	
Average		Average	79.44	95.56	90.27
		Maximum	79.40	95.55	90.09
		Stacking	80.62	96.01	90.09
		Late Fusion	79.01	95.49	90.25
		LFC	81.13	96.14	91.32
		CMoDE	81.40	96.12	90.29
		SSMA (ours)	82.64	96.41	90.65
	SSMA_msf (ours)	<b>83.94</b>	<b>96.68</b>	<b>91.99</b>	

All the fusion models have the same unimodal AdapNet++ network backbone. Results on the test set are shown in Table 2

Bold numbers indicate the best performance among the compared methods

fusion achieves a mIoU of 83.29% outperforming all the other approaches and setting the new state-of-the-art. The SSMA\_msf model further improves upon the performance of the SMMA model by 1.3%. As the Cityscapes dataset does not contain harsh environments, the improvement that can be achieved using fusion is limited to scenes that contain inconspicuous object classes or mismatched relationship. However, the additional robustness that it demonstrates due to multimodal fusion is still notable as shown in the qualitative results in the following sections. Additionally, the benchmarking results on the Cityscapes test set is shown in Table 2. The results demonstrate that our SSMA fusion architecture with the AdapNet++ network backbone achieves a comparable performance as the top performing DPC and DRN architectures, while outperforming the other networks on the leaderboard.

We benchmark on the Synthia dataset to demonstrate the utility of fusion when both modalities contain rich information. It consists of scenes with adverse perceptual conditions including rain, snow, fog and night, therefore the benefit of multimodal fusion for outdoor environments is most evident on this dataset. As the Synthia dataset does not provide camera calibration parameters, we cannot compute the HHA encoding, therefore we benchmark using visual RGB and depth images. Results from benchmarking on this dataset are

**Table 18** Comparison of multimodal fusion approaches on the Synthia validation set (input image dim:  $768 \times 384$ )

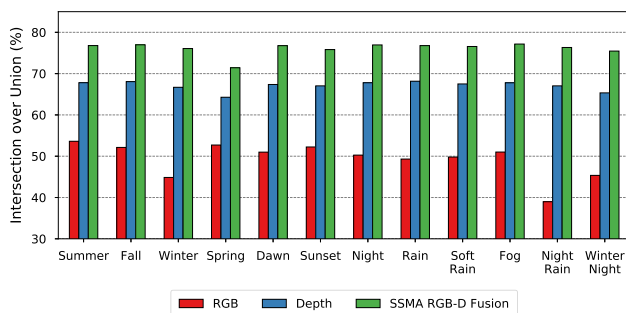
Network	Approach	mIoU (%)	Acc. (%)	AP (%)
RGB	Unimodal	86.70	97.18	93.17
Depth	Unimodal	87.87	97.78	94.23
RGB-D	Average	89.22	98.03	95.04
	Maximum	89.13	98.01	94.97
	Stacking	88.95	98.03	94.41
	Late fusion	89.13	98.01	94.66
	LFC	89.48	98.09	94.96
	CMoDE	89.57	98.13	94.58
	FuseNet	89.62	98.34	94.67
	SSMA (ours)	91.25	98.48	95.68
SSMA_msf (ours)	<b>92.10</b>	<b>98.64</b>	<b>96.37</b>	

All the fusion models have the same unimodal AdapNet++ network backbone

Bold numbers indicate the best performance among the compared methods

shown in Table 18. Due to the high-resolution depth information, the unimodal depth model achieves a mIoU of 87.87%, outperforming segmentation using visual RGB images by 1.17%. This demonstrates that accurate segmentation can be obtained using only depth images as input provided that the





**Fig. 19** Evaluation of our proposed SSMA fusion technique on the Synthia-Sequences dataset containing a variety of seasons and weather conditions. We use the models trained on the Synthia-Rand-Cityscapes dataset and only test on the individual conditions in the Synthia-Sequences dataset to quantify its robustness. Our model that performs RGB-D fusion consistently outperforms the unimodal models which can be more prominently seen qualitatively in Fig. 21c, d (Color figure online)

depth sensor gives accurate long range information. Among the baseline fusion approaches and the state-of-the-art techniques, the CMoDE RGB-D fusion approach achieves the highest mIoU, outperforming the unimodal depth model by 1.7%. While our proposed SSMA architecture demonstrates state-of-the-art performance of 91.25% and further improves the mIoU to 92.10% using the SSMA\_msf model. This accounts to a large improvement of 5.4% over the best performing unimodal segmentation model. Other metrics such as the pixel accuracy and average precision also show similar improvement.

One of our main motivations to benchmark on this dataset is to evaluate our SSMA fusion model on a diverse set of scenes with adverse perpetual conditions. For this experiment, we trained our SSMA fusion model on the Synthia-Rand-Cityscapes training set and evaluated the performance on each of the conditions contained in the Synthia-Sequences dataset. The Synthia-Sequences dataset contains individual video sequences in different conditions such as *summer*, *fall*, *winter*, *spring*, *dawn*, *sunset*, *night*, *rain*, *soft rain*, *fog*, *night rain* and *winter night*. Results from this experiment are shown in Fig. 19. The unimodal visual RGB model achieves an overall mIoU score of  $49.27\% \pm 4.04\%$  across the 12 sequences. Whereas, the model trained on the depth maps achieves a mIoU score of  $67.07\% \pm 1.12\%$ , thereby substantially outperforming the model trained using visual RGB images.

As this is a synthetic dataset, the depth maps provided are accurate and dense even for structures that are several hundreds of meters away from the camera. Therefore, this enables the unimodal depth model to learn representations that accurately encode the structure of the scene and these structural representations are proven to be invariant to the change in perceptual conditions. It can also be observed that the unimodal depth model performs consistently well in all the conditions with a variance of 1.24%, demonstrating the

generalization to different conditions. However, the visual RGB model with a variance of 16.30% performs inconsistently across different conditions. Nevertheless, we observe that our RGB-D SSMA fusion model outperforms the unimodal visual RGB model by achieving a mIoU score of  $76.51\% \pm 0.49\%$  across the 12 conditions, accounting to an improvement of 27.24%. Moreover, the SSMA fusion model has a variance of 0.24%, demonstrating better generalization ability across varying adverse perceptual conditions.

We also benchmark on the indoor SUN RGB-D dataset which poses a different set of challenges than the outdoor datasets. The improvement from multimodal fusion is more evident here as indoor scenes are often smaller confined spaces with several cluttered objects and the depth modality provides valuable structural information that can be exploited. Results from RGB-D and RGB-HHA multimodal fusion is shown in Table 19. Among the individual modalities, segmentation using visual RGB images yields the highest mIoU of 38.40%. The model trained on depth images performs 4.13% lower than the visual RGB model. This can be attributed to the fact that the depth images are extremely noisy with numerous missing depth values in the SUN RGB-D dataset. However, our proposed SSMA fusion on RGB-HHA achieves state-of-the-art performance with a mIoU of 44.43%, constituting a substantial improvement of 6.03% over the unimodal visual RGB model. Moreover, our SSMA\_msf model further improves upon the mIoU by 1.3%. Similar to the performance in other datasets, RGB-HHA achieves a higher mIoU than RGB-D fusion corroborating the fact that CNNs learn more effectively from the HHA encoding but with a small additional preprocessing time.

Table 20 shows the results on the ScanNet validation set. ScanNet is the largest indoor RGB-D dataset to date with over 1513 different scenes and 2.5M views. Unlike the SUN RGB-D dataset, ScanNet contains depth maps of better quality with lesser number of missing depth values. The unimodal visual RGB model achieves an mIoU of 52.92% with a pixel accuracy of 77.70%, while the unimodal HHA model achieves an mIoU of 54.19% with an accuracy of 80.20%. For multimodal fusion, CMoDE using RGB-HAA demonstrates the highest performance among state-of-the-art architectures achieving a mIoU of 64.07%. While our proposed SSMA RGB-HHA model outperforms CMoDE by yielding a mIoU of 66.34%, which is a significant improvement of 13.42% over the unimodal visual RGB model. Moreover, the SSMA\_msf model further improves the mIoU score to 67.52%. To the best of our knowledge, this is the largest improvement due to multimodal fusion obtained thus far. An interesting observation that can be made from the results on SUN RGB-D and ScanNet is that the lowest multimodal fusion performance is obtained using the Stacking fusion approach, reaffirming our hypothesis that fusing more semantically mature features enables the model to exploit

**Table 19** Comparison of multimodal fusion approaches on the SUN RGB-D validation set (input image dim:  $768 \times 384$ )

Network	Approach	mIoU (%)	Acc. (%)	AP (%)
RGB	Unimodal	38.40	76.90	62.78
Depth	Unimodal	34.27	73.83	74.39
HHA	Unimodal	34.59	74.39	57.18
RGB-D	Average	40.70	78.58	64.54
	Maximum	40.58	78.50	64.04
	Stacking	36.48	76.68	57.92
	Late Fusion	41.68	79.27	66.63
	LFC	41.82	79.36	66.75
	CMoDE	41.87	79.84	66.81
	FuseNet	41.85	79.56	66.87
	SSMA (ours)	43.90	80.16	66.11
	SSMA_msf (ours)	<b>44.52</b>	<b>80.67</b>	<b>67.92</b>
	RGB-HHA	Average	41.01	78.54
Maximum		40.91	78.49	64.78
Stacking		37.49	76.42	57.88
Late fusion		41.91	79.49	67.31
LFC		42.42	79.55	67.41
CMoDE		42.55	79.94	65.38
SSMA (ours)		44.43	80.21	64.94
SSMA_msf (ours)		<b>45.73</b>	<b>80.97</b>	<b>67.82</b>

All the fusion models have the same unimodal AdapNet++ network backbone  
 Bold numbers indicate the best performance among the compared methods

**Table 20** Comparison of multimodal fusion approaches on the ScanNet validation set (input image dim:  $768 \times 384$ )

Network	Approach	mIoU (%)	Acc. (%)	AP (%)
RGB	Unimodal	52.92	77.70	77.28
Depth	Unimodal	53.80	80.63	74.46
HHA	Unimodal	54.19	80.20	73.90
RGB-D	Average	58.20	82.31	79.45
	Maximum	57.68	82.12	78.35
	Stacking	55.89	79.04	77.08
	Late fusion	61.37	82.48	80.15
	LFC	62.97	83.70	80.93
	CMoDE	64.00	84.94	81.27
	FuseNet	63.83	84.24	81.02
	SSMA (ours)	66.29	86.11	81.81
	SSMA_msf (ours)	<b>67.38</b>	<b>86.24</b>	<b>81.93</b>
	RGB-HHA	Average	58.39	82.36
Maximum		57.88	82.12	77.34
Stacking		56.48	80.33	77.47
Late fusion		61.45	82.68	80.12
LFC		63.09	83.67	80.67
CMoDE		64.07	84.84	80.80
SSMA (ours)		66.34	86.02	81.49
SSMA_msf (ours)		<b>67.52</b>	<b>86.38</b>	<b>82.21</b>

All the fusion models have the same unimodal AdapNet++ network backbone. Results on the test set are shown in Table 6  
 Bold numbers indicate the best performance among the compared methods

complementary properties from the modalities more effectively. We also benchmark on the ScanNet test set and report the results in Table 6. Our proposed SSMA fusion architecture with the AdapNet++ network backbone sets the new state-of-the-art on the ScanNet benchmark.

Finally, we present benchmarking results on the Freiburg Forest dataset that contains three inherently different modalities including visual RGB images, Depth data and EVI. EVI or Enhanced Vegetation Index was designed to enhance the vegetation signal in high biomass regions and it is computed from the information contained in three bands, namely, Near-Infrared, Red and Blue channels (Running et al. 1999). As this dataset contains scenes in unstructured forested environments, EVI provides valuable information to discern between inconspicuous classes such as *vegetation* and *grass*. Table 21 shows the results on this dataset for multimodal fusion of RGB-D and RGB-EVI. For unimodal segmentation, the RGB model yields the highest performance, closely followed by the model trained on EVI. While for multimodal segmentation, our SSMA model trained on RGB-EVI yields the highest mIoU of 83.90% and our SMMA\_msf model further improves upon the performance and achieves a mIoU of 84.18%. Both these models outperform existing multimodal fusion methods and set the new state-of-the-art.

## 5.8 Multimodal Fusion Discussion

To summarize, the models trained on visual RGB images perform the best in general in comparison to unimodal segmentation with other modalities. However, when the depth data is less noisy and the environment is a confined indoor space, the model trained on depth or HHA encoded depth outperforms visual RGB models. Among the multimodal fusion baselines, late fusion and Stacking, each perform well in different environments. Stacking performs better in outdoor environments, while late fusion performs better indoors. This can be attributed to the fact that the late fusion method fuses semantically mature representations. Therefore, in indoor environments, modalities such as depth maps from stereo cameras are less noisy than in outdoors and as the environment is confined, all the objects in the scene are well represented with dense depth values. This enables the late fusion architecture to leverage semantically rich representations for fusion. However in outdoor environments, depth values are very noisy and no information is present for objects at far away distances. Therefore, the semantic representations from the depth stream are considerably less informative for certain parts of the scene which does not allow the Late Fusion network to fully exploit complementary features and hence it does not provide significant gains. On the other hand, in indoor or synthetic scenes where the depth modality is dense and rich with information, late fusion generally outperforms the stacking approach.

Among the current state-of-the-art methods, CMoDE outperforms all other approaches in most of the diverse environments. To recapitulate, CMoDE employs a class-wise probabilistic late fusion technique that adaptively weighs the modalities based on the scene condition. However, our proposed SSMA fusion techniques outperforms CMoDE in all the datasets and sets the new state-of-the-art in multimodal semantic segmentation. This demonstrates that fusion of modalities is an inherently complex problem that depends on several factors such as the object class of interest, the spatial location of the object and the environmental scene context. Our proposed SSMA fusion approach dynamically adapts the fusion of semantically mature representations based on the aforementioned factors, thereby enabling our model to effectively exploit complementary properties from the modalities. Moreover, as the dynamicity is learned in a self-supervised fashion, it efficiently generalizes to different diverse environments, perceptual conditions and the types of modalities employed for fusion. Another advantage of this dynamic adaptation property of our multimodal SSMA fusion mechanism is the intrinsic tolerance to sensor failure. In case one of the modalities becomes unavailable, the SSMA module can be trained to switch the output of the unavailable modality-specific encoder off by generating gating probabilities as zeros. This enables the multimodal model to still yield a valid segmentation output using only the modality that is available and the performance is comparable to that of the unimodal model with the remaining modality.

## 5.9 Generalization of SSMA Fusion to Other Tasks

In order to demonstrate the generalization of our proposed SSMA module for multimodal fusion in other tasks, we report results for the scene type classification task on the ScanNet benchmark. The goal of the scene type classification task is to classify scans of indoor scenes into 13 distinct categories, namely, *apartment*, *bathroom*, *bedroom/hotel*, *bookstore/library*, *conference room*, *copy/mail room*, *hallway*, *kitchen*, *laundry room*, *living room/lounge*, *misc*, *office*, *storage/basement/garage*. The benchmark ranks the methods according to recall and the intersection-over-union metric (IoU). For our approach, we employ the top-down 2D projection of the textured scans as one modality and the jet-colored depth map of the top-down 2D projection of the scans as another modality. We utilize the SE-ResNetXt-101 (Hu et al. 2018) architecture for the unimodal model and for the multimodal network backbone. Our multimodal architecture has a late fusion topology with two individual modality-specific SE-ResNetXt-101 streams that are fused after block 5 using our SSMA module. The output of the SSMA module is fed to a fully connected layer that has number of output units equal to the number of scene classes in the dataset. We evaluate the performance of our multimodal

**Table 21** Comparison of multimodal fusion approaches on the Freiburg Forest validation set (input image dim:  $768 \times 384$ )

Network	Approach	mIoU (%)	Acc. (%)	AP (%)
RGB-D	RGB	83.09	95.15	89.83
	Depth	73.93	91.42	85.36
	EVI	80.96	94.20	88.88
	Average	79.51	92.87	90.99
	Maximum	81.62	93.93	90.87
	Stacking	83.13	95.19	89.95
	Late Fusion	82.11	93.95	90.85
	LFC	82.53	94.99	90.96
	CMoDE	83.21	95.19	90.19
	FuseNet	83.10	95.07	90.03
	SSMA (ours)	83.81	95.62	92.78
	SSMA_msf (ours)	<b>83.99</b>	<b>95.70</b>	<b>93.08</b>
	RGB-EVI	Average	83.00	95.10
Maximum		83.00	95.10	90.17
Stacking		83.18	95.21	90.11
Late fusion		82.80	95.01	90.07
LFC		83.00	95.13	90.28
CMoDE		83.31	95.22	90.19
SSMA (ours)		83.90	95.56	92.28
SSMA_msf (ours)		<b>84.18</b>	<b>95.64</b>	<b>92.60</b>

All the fusion models have the same unimodal AdapNet++ network backbone

**Table 22** Performance of multimodal SSMA fusion for the scene type classification task on the ScanNet benchmark

Network	Approach	mIoU (%)	mRecall (%)
RGB	Unimodal	33.16	43.37
Depth	Unimodal	35.07	44.82
RGB-D	Average	34.92	43.99
	Maximum	34.75	43.79
	Stacking	32.47	41.89
	Late fusion	35.17	44.48
	SSMA (ours)	<b>37.45</b>	<b>54.28</b>

Results are reported on the validation set  
Bold numbers indicate the best performance among the compared methods

SSMA fusion model against the individual modality-specific networks, as well as the multimodal fusion baselines such as Average, Maximum, Stacking and Late Fusion, as described in Sect. 5.7.

Results from this experiment on the ScanNet validation set are shown in Table 22. It can be seen that the unimodal depth model outperforms the RGB model in both the mean IoU (mIoU) score and the mean recall (mRecall). Among the multimodal fusion baselines, only the Late Fusion network outperforms the unimodal depth model by a small margin in the mIoU score but it achieves a lower mean recall. However, our multimodal SSMA fusion model achieves the state-of-the-art performance with a mIoU score of 37.45% and a

mean recall of 54.28%. This accounts for an improvement of 2.28% in the mIoU score and 9.8% in the mean recall over the Late Fusion model, and a larger improvement over the performance of the unimodal depth model. Since the only difference in the Late Fusion architecture and the SSMA architecture is how the multimodal fusion is carried out, the improvement achieved by the SSMA model can be solely attributed to the dynamic fusion mechanism of our SSMA module. We also benchmarked on the ScanNet test set for scene classification, in which our multimodal SSMA model achieves a mIoU score of 35.5% and a mean recall of 49.8%, thereby setting the state-of-the-art for scene type classification on this benchmark.

## 5.10 Multimodal Fusion Ablation Studies

In this section, we study the influence of various contributions that we make for multimodal fusion. Specifically, we evaluate the performance by comparing the fusion at different intermediate network stages. We then evaluate the utility of our proposed channel attention scheme for better correlation of mid-level encoder and high-level decoder features. Subsequently, we experiment with different SSMA bottleneck downsampling rates and qualitatively analyze the convolution activation maps of our fusion model at various intermediate network stages to study the effect of multimodal fusion on the learned network representations.



**Table 23** Performance of multimodal SSMA fusion technique with different real-time backbone networks

Network backbone	Modalities	mIoU (%)	Time (ms)
ERFnet	RGB	62.71	43.87
	HHA	51.84	43.87
	RGB-HHA SSMA	64.60	66.06
MobileNet v2	RGB	74.78	47.43
	HHA	61.89	47.43
	RGB-HHA SSMA	77.17	73.62
AdapNet++	RGB	80.80	72.77
	HHA	67.66	72.77
	RGB-HHA SSMA	82.64	99.96

Results are shown on the Cityscapes validation set (input image dim:  $768 \times 384$ )

**Table 24** Effect of the various contributions proposed for multimodal fusion using AdapNet++ and the SMMA architecture

Model	SSMA Fusion			mIoU (%)	Acc. (%)	AP (%)
	ASPP	Skip	Ch. Agg.			
F0	–	–	–	80.77	96.04	90.97
F1	✓	–	–	81.55	96.19	91.15
F2	✓	✓	–	81.75	96.25	91.09
F3	✓	✓	✓	<b>82.64</b>	<b>96.41</b>	<b>90.65</b>

The performance is shown for RGB-HHA fusion on the Cityscapes dataset and evaluated on the validation set

Bold numbers indicate the best performance among the compared methods

### 5.10.1 Experiments with Real-Time Backbone Networks

In the interest of real-time performance, we additionally trained multimodal models in our proposed SSMA fusion framework with different real-time-intended backbone networks. Specifically, we performed experiments using two different backbone networks: ERFnet (Romera et al. 2018) and MobileNet v2 (Sandler et al. 2018). For the ERFnet fusion model, we replace the two modality-specific encoders in our multimodal fusion configuration with the ERFnet encoder and we replace our decoder with the ERFnet decoder. While for the fusion model with the MobileNet v2 backbone, we employ the MobileNet v2 topology for the two modality-specific encoders in our multimodal fusion configuration and we append our eASPP as well as the decoder from our AdapNet++ architecture. Note that ERFnet is a semantic segmentation architecture with both an encoder and decoder, while MobileNet v2 is only a classification architecture and therefore it only has an encoder topology.

Results from this experiment for RGB-HHA fusion along with the unimodal RGB and unimodal HHA performance are shown in Table 23. The ERFnet model in our multimodal SSMA fusion configuration achieves a mIoU score of 64.60% with an inference time of 66.06ms, thereby outperforming both the unimodal RGB ERFnet model and the unimodal HHA ERFnet model. The MobileNet v2 model in our multimodal SSMA fusion configuration further outperforms the ERFnet fusion model by 12.57% in the mIoU score

with an inference time of 73.62ms. In comparison to these network backbones, our AdapNet++ fusion model achieves a mIoU score of 82.84% with an inference time of 99.96ms. Each of these multimodal fusion models outperform their unimodal counterparts and have an inference time in the range of 66ms – 99ms. This demonstrates the modularity of our fusion framework that enables the selection of an appropriate network backbone according to the desired frame rate.

### 5.10.2 Detailed Study on the Fusion Architecture

In our proposed multimodal fusion architecture, we employ a combination of mid-level fusion and late-fusion. Results from fusion at each of these stages is shown in Table 24. First, we employ the main SSMA fusion module at the end of the two modality-specific encoders, after the eASPPs and we denote this model as F1. The F1 model achieves a mIoU of 81.55%, which constitutes to an improvement of 0.78% over the unimodal F0 model. We then employ an SSMA module at each skip refinement stage to fuse the mid-level skip features from each modality-specific stream. The fused skip features are then integrated into the decoder for refinement of high-level decoder features. The F2 model that performs multimodal fusion at both stages, yields a mIoU of 81.75%, which is not significant compared to the improvement that we achieve in fusion of the mid-level features into the decoder in our unimodal AdapNet++ architecture. As described in Sect. 4.2, we hypothesise that this is due to the fact that the

**Table 25** Effect of varying the SSMA bottleneck downsampling rate  $\eta$  on the RGB-HHA fusion performance

Encoder SSMA	Skip SSMA	mIoU (%)	Acc. (%)	AP (%)
$\eta_{enc} = 2$	$\eta_{skip} = 2$	82.15	96.32	91.17
$\eta_{enc} = 4$	$\eta_{skip} = 4$	82.11	96.27	91.34
$\eta_{enc} = 8$	$\eta_{skip} = 4$	82.21	96.32	91.61
$\eta_{enc} = 16$	$\eta_{skip} = 4$	82.25	96.31	91.12
$\eta_{enc} = 16$	$\eta_{skip} = 6$	<b>82.64</b>	<b>96.41</b>	<b>90.65</b>

Results are shown for the model trained on the Cityscapes dataset and evaluated on the validation set

Bold numbers indicate the best performance among the compared methods

mid-level representations learned by a network do not align across different modality-specific streams. Therefore, we employ our channel attention mechanism to better correlate these features using the spatially aggregated statistics of the high-level decoder features. The model that incorporates this proposed attention mechanism achieves an improved mIoU of 82.64%, which is an improvement of 1.09% in comparison to 0.2% without the channel attention mechanism. Note that the increase in quantitative performance due to multimodal fusion is more apparent in the indoor or synthetic datasets as shown in Sect. 5.7. The experiment shown in Table 24 shows a correspondingly larger increase for the contributions presented in this table. However, as we present the unimodal ablation studies on the Cityscapes dataset, we continue to show the multimodal ablation studies on the same dataset.

The proposed SSMA module has a bottleneck structure in which the middle convolution layer downsamples the number of feature channels according to a rate  $\eta$  as described in Sect. 4.1. As we perform fusion both at the mid-level and at the end of the encoder section, we have to estimate the downsampling rates individually for each of the SSMA blocks. We start by using values from a geometric sequence for the main encoder SSMA downsampling rate  $\eta_{enc}$  and correspondingly vary the values for the skip SSMA downsampling rates  $\eta_{skip}$ . Results from this experiment shown in Table 25 demonstrates that the best performance is obtained for  $\eta_{enc} = 16$  and  $\eta_{skip} = 6$  which also increases the parameter efficiency compared to lower downsampling rates.

### 5.10.3 Fusion Stage and Reliance on Modalities

In this section, we study the SSMA fusion configuration in terms of learning dependent or independent probability weightings that are used to recalibrate the modality-specific feature maps dynamically. The SSMA configuration that we depict in Fig. 7 learns independent probability weighting where the activations at a specific location in the feature maps from modality A can be independently enhanced or suppressed regardless of whether the activations at the cor-

responding location in the feature maps from modality B are going to be enhanced or suppressed. An alternative dependent configuration can be employed by replacing the sigmoid with a softmax, where the softmax takes the activation at a specific location from a feature map from modality A and the activation in the corresponding location from the feature map from modality B, and outputs dependent probabilities that are used to weigh the modality-specific activations. This dependent configuration acts as punishing the modality that makes the mistake while rewarding the other. While the independent configuration also considers if a modality is making a mistake, it does not necessarily punish one and reward the other, it has the ability to punish both or reward both in addition. We study the performance of the multimodal fusion in these two dependent and independent configurations in this section. Additionally, we study the effect of learning the fusion in a fully supervised manner by employing an explicit loss function at the output of each SSMA module after the fusion. We also study the overall configuration on where the SSMA module is to be placed, at the end of the encoder stage where the features are highly discriminative or at the end of the decoder stage where the features are more high-level and semantically mature.

Table 26 shows the results from this experiment where we present the multimodal fusion performance for the model in the dependent SSMA and independent SSMA configuration, as well as when the SSMA module is placed at the encoder stages as in our standard configuration or at the decoder stage, and with and without an explicit supervision for the fusion. The results are presented for RGB-HHA fusion on the Cityscapes validation set. It can be seen that the models S1 to S4 without the explicit supervision outperform the corresponding models S5 to S8 with the explicit supervision demonstrating that learning the fusion in self-supervised manner is more beneficial. Comparing the performance of models that employ the SSMA fusion at the encoder stages S1, S2, S5, S6, with the corresponding models at employ the SSMA fusion at the decoder stage S3, S4, S7, S8, we see that the encoder fusion models substantially outperform the decoder fusion models. Finally, comparing the models with the dependent SSMA configuration S1, S3, S5, S7 with the corresponding models the independent SSMA configuration, we observe that independent configuration always outperforms the dependent configuration. In summary, these results demonstrate that our multimodal SSMA fusion scheme that learns independent probability weightings in a self-supervised manner, when employed at the encoder stages outperforms the other configurations.

### 5.10.4 Influence of Multimodal Fusion on Activation Maps

In an effort to present visual explanations for the improvement in performance due to multimodal fusion, we study

**Table 26** Comparison of multimodal SSMA fusion at different network stages and by employing a dynamic dependent probability weighting or independent probability weighting configuration

Model	SSMA loss	Encoder Stage		Decoder Stage		IoU (%)
		Dep.	Indep.	Dep.	Indep.	
S1	–	✓	–	–	–	82.39
S2	–	–	✓	–	–	<b>82.64</b>
S3	–	–	–	✓	–	80.96
S4	–	–	–	–	✓	81.14
S5	✓	✓	–	–	–	82.32
S6	✓	–	✓	–	–	82.52
S7	✓	–	–	✓	–	80.86
S8	✓	–	–	–	✓	81.06

Results are shown for RGB-HHA fusion on the Cityscapes validation set with (supervised) and without (self-supervised) an explicit loss for the SSMA module

Bold number denote the highest performance among the compared methods

the activation maps at various intermediate network stages before and after the multimodal fusion using the GradCam++ technique (Chattopadhyay et al. 2017). The approach introduces pixel-wise weighting of the gradients of the output with respect to a particular spatial location in the convolutional feature map to generate a score. The score provides a measure of the importance of each location in feature map towards the overall prediction of the network. We apply a colormap over the obtained scores to generate a heat map as shown in Fig. 20. We visualize the activation maps at five different stages of the network. Firstly, at the output of each modality-specific encoder  $X^a$  and  $X^b$  which is the input to the SSMA fusion block. Secondly, after recalibrating the individual modality-specific feature maps inside the SSMA block  $\hat{X}^a$  and  $\hat{X}^b$ , and finally after the fusion with the  $3 \times 3$  convolution inside the SSMA block  $f$ . Figure 20 illustrates one example for each dataset that we benchmark on with the activation maps, the input modalities and the corresponding segmentation output for the particular object category.

For the Cityscapes dataset, we show the activation maps for the *person* category. It can be seen that the activation map  $X^a$  from the visual RGB stream is well defined for the *person* class but it does not show high activations centered on the objects, whereas the activation map from the depth stream  $X^b$  is more noisy but high activations are shown on the objects. For the locations in the input depth map that show noisy depth data, the activation map correspondingly shows prominent activations in these regions. After the recalibration of the feature maps, both  $\hat{X}^a$  and  $\hat{X}^b$  are less noisy and maintaining the structure with high activations. Furthermore, the activation map of the fused convolution  $f$  shows very well defined high activations that almost correspond to the segmented output.

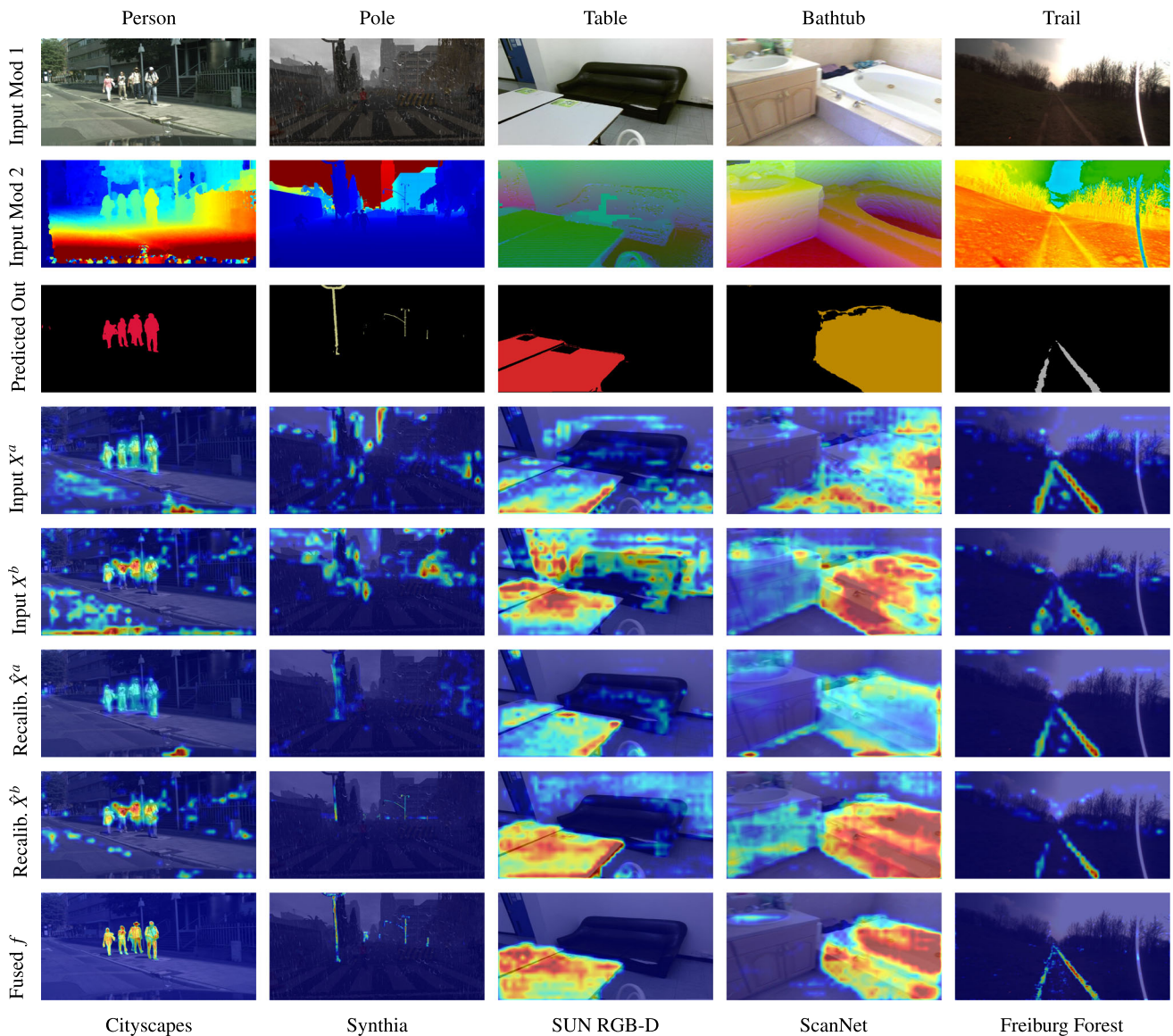
The second column in Fig. 20 shows the activations for the *pole* class in the Synthia dataset. As the scene was captured during rainfall, the objects in the visual RGB image are indistinguishable. However, the depth map still maintains some

structure of the scene. Studying both the modality-specific activation maps at the input to the SSMA module shows substantial amount of noisy activations spread over the scene. Therefore, the unimodal visual RGB model only achieves an IoU of 74.94% for the pole class. Whereas, after the recalibration of the feature maps, the activation maps show significantly reduced noise. It can be seen the recalibrated activation map  $\hat{X}^b$  of the depth stream shows more defined high activations on the pole, whereas  $\hat{X}^a$  of the visual RGB stream shows less amount of activations indicating that the network suppresses the noisy RGB activations in order to better leverage the well defined features from the depth stream. Activations of the final fused convolution layer show higher activations on the pole than either of the recalibrated activation maps demonstrating the utility of multimodal fusion. This enables the fusion model to achieve an improvement of 8.4% for the pole class.

The third column of Fig. 20 shows the activation maps for the *table* class in the SUN RGB-D dataset. Interestingly, both the modality-specific activation maps at the input show high activations at different locations indicating the complementary nature of the features in this particular scene. However, the activation map  $X^b$  from the HHA stream also shows high activations on the couch in the background which would cause misclassifications. After the recalibration of the HHA feature maps, the activation map  $\hat{X}^b$  no longer has high activations on the couch but it retains the high activations on the table. While, the recalibrated activation map  $\hat{X}^a$  of the visual RGB stream shows significantly lesser noisy activations. The activation map of the fused convolution  $f$  shows well defined high activations on the table, more than the modality-specific input activation maps. The enables the SSMA fusion model to achieve an improvement of 4.32% in the IoU for the table category.

For the ScanNet dataset, we show the activation maps for the *bathtub* category in the fourth column in Fig. 20. It can be seen that the modality specific activation maps at the input





**Fig. 20** Visualization of the activation maps with respect to a particular class at various stages of the network before and after multimodal fusion.  $X^a$  and  $X^b$  are at the outputs of the modality-specific encoder which is input to the SSMA fusion module,  $\hat{X}^a$  and  $\hat{X}^b$  are the at the

feature maps after recalibration inside the SSMA block, and  $f$  is after the fusion of both modalities inside the SSMA block. Both the input modalities and the corresponding segmentation output for the particular object category is also shown (Color figure online)

of the SSMA module shows high activations at complementary locations, corroborating the utility of exploiting features from both modalities. Moreover, the activation map  $X^b$  from the HHA stream shows significantly higher activations on the object of interest than the RGB stream. This also aligns with the quantitative results, where the unimodal HHA model outperforms the model trained on visual RGB images. After the recalibration of the feature maps inside the SSMA block, the activation maps show considerably lesser noise while maintaining the high activations at complementary locations. The activation map of the fused convolution  $f$  shows only high

activations on the bathtub and resembles the actual structure of the segmented output.

The last column of Fig. 20 shows the activation maps for the *trail* category in the Freiburg Forest dataset. Here we show the fusion with visual RGB and EVI. The EVI modality does not provide substantial complementary information for the trail class in comparison to the RGB images. This is also evident in the visualization of the activations at the input of the SSMA module. The activation maps of the EVI modality after the recalibration show significantly lesser noise but also lesser amount of high activation regions than the recalibrated activation maps of the visual RGB stream. Nevertheless, the



activation map after the fusion  $f$  shows more defined structure of the trail than either of the modality-specific activation maps of the input to the SSMA module.

### 5.11 Qualitative Results of Multimodal Segmentation

Figure 21 illustrates the visualized comparisons of multimodal semantic segmentation for each of the five benchmark datasets. We compare with the output of unimodal AdapNet++ architecture and show the improvement/error map which denotes the improvement over the unimodal AdapNet++ output in green and the misclassifications in red. Figure 21a, b show interesting examples from the Cityscapes dataset. In both these examples, we can see a significant improvement in the segmentation of *cyclists*. As cyclists constitute to a person riding a bike, often models assign a part of the pixels on the person riding a bike as the person class, instead of the *cyclist* class.

Another common scenario is when there is a person standing a few meters behind a parked bike, the model misclassifies the *person* as a *cyclist* but since he is not on the bike, the right classification would be the person category. In these examples, we can see that by leveraging the features from the depth modality our network makes accurate predictions in these situations. In Fig. 21a, we can also see that parts of the car at several meters away is not completely segmented in the unimodal output but it is accurately captured in the multimodal output. Furthermore, in the unimodal output of Fig. 21b, we see parts of the *sidewalk* behind the people is misclassified as *road* and parts of the *fence* that is several meters away is misclassified as a *sidewalk*. As the distinction between these object categories can clearly be seen in the depth images, our fusion model accurately identifies these boundaries.

Figure 21c, d show examples on the Synthia dataset. Here we show the first scene during rainfall and the second scene during night-time. In the unimodal output of first scene, we can see significant misclassifications in all the object categories, except *building* and *vegetation* that are substantially large in size. Whereas, the multimodal fusion model is able to leverage the depth features to identify the objects in the scene. In Fig. 21d, even for us humans it is impossible to see the *people* on the road due to the darkness in the scene. As predicted, the unimodal visual RGB model misclassifies the entire road with people as a *car*, which could lead to disastrous situations if it occurred in the real-world. The multimodal model is able to accurately predict the scene with almost no error in the predictions.

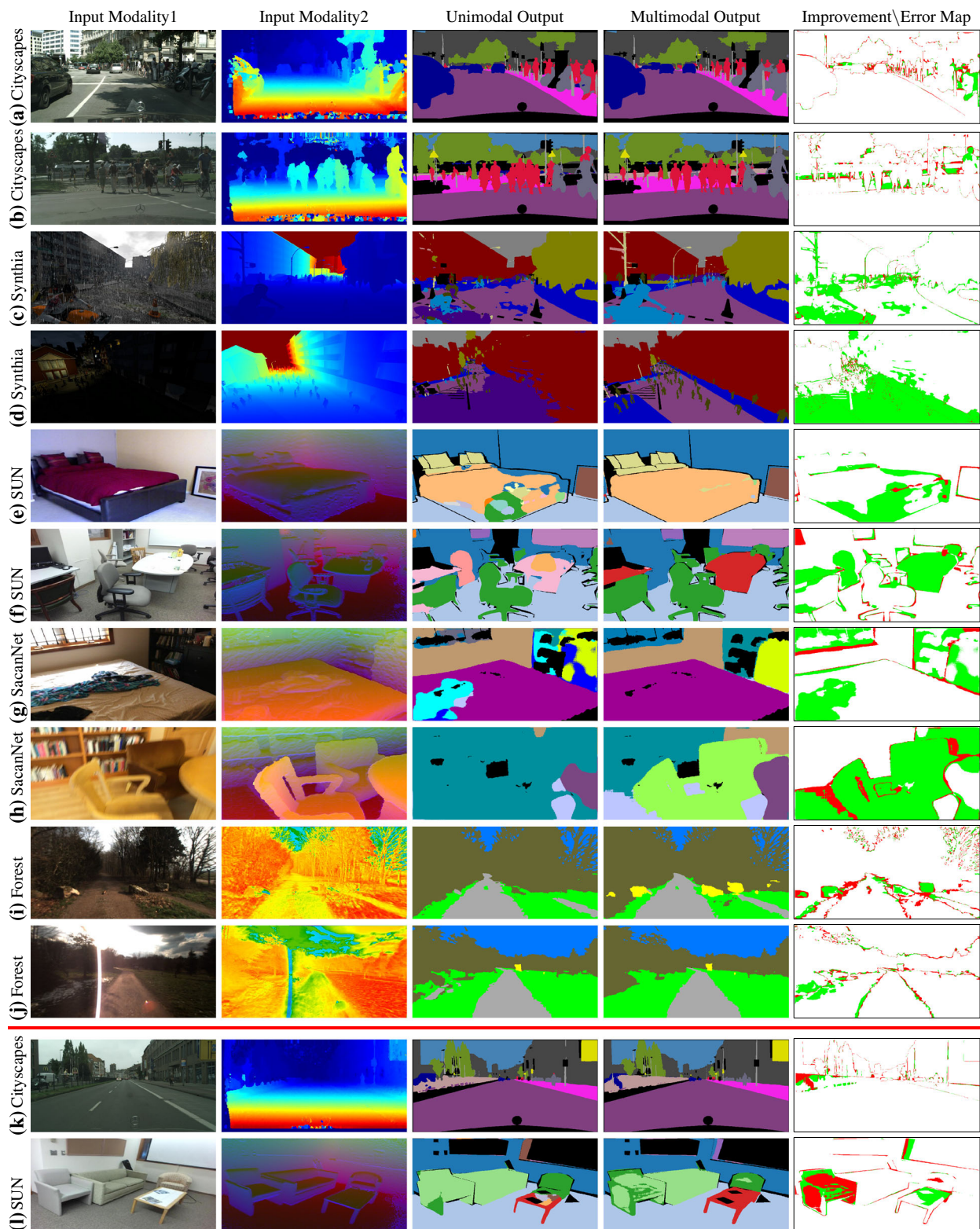
In Fig. 21e, f, we show examples on the indoor SUN RGB-D dataset. Due to the large number of object categories in this dataset, several inconspicuous classes exist. Leveraging structural properties of objects from the HHA encoded depth can enable better discrimination between them. Figure 21e

shows a scene where the unimodal model misclassifies parts of the wooden *bed* as a *chair* and parts of the *pillow* as the *bed*. We can see that the multimodal output significantly improves upon the unimodal counterpart. Figure 21f shows a complex indoor scene with substantial clutter. The unimodal model misclassifies the *table* as a *desk* and a hatch in the wall is misclassified as a *door*. Moreover, partly occluded *chairs* are not entirely segmented in the unimodal output. The HHA encoded depth shows well defined structure of these objects, which enables the fusion approach to precisely segment the scene. Note that the *window* in the top left corner of Fig. 21f is mislabeled as a *desk* in the groundtruth.

Figure 21g, h show examples of indoor scenes from the ScanNet dataset. In the unimodal output of Fig. 21g, overexposure of the image near the windows causes misclassification of parts of the *window* as a *picture* and crumpled bedsheets as well as the *bookshelf* is missclassified as a *desk*. While the multimodal segmentation output does not demonstrate these errors. Figure 21h shows an image with motion-blur due to camera motion. The motion blur causes a significant percentage of the image to be misclassified as the largest object in the scene and this case as a *bookshelf*. Analyzing the HHA encoded depth map, we can see that it does not contain overexposed sections or motion-blur, rather it strongly emphasizes the structure of the objects in the scene. By leveraging features from the HHA encoded depth stream, our multimodal model is able to accurately predict the object classes in the presence of these perceptual disturbances.

In Fig. 21i, j, we show results on the unstructured Freiburg Forest dataset. Figure 21i shows an oversaturated image due to sunlight which causes boulders on the *grass* to be completely absent in the unimodal segmentation output. Oversaturation causes boulders to appear with a similar texture as the *trail* or *vegetation* class. However, the RGB-EVI multimodal model is able to leverage the complementary EVI features to segment these structures. Figure 21j shows an example scene with glare on the camera optics and snow on the ground. In the unimodal semantic segmentation output, the presence of these disturbances often causes localized misclassifications in the areas where they are present. Whereas, the multimodal semantic segmentation model compensates for these disturbances exploiting the complementary modalities.

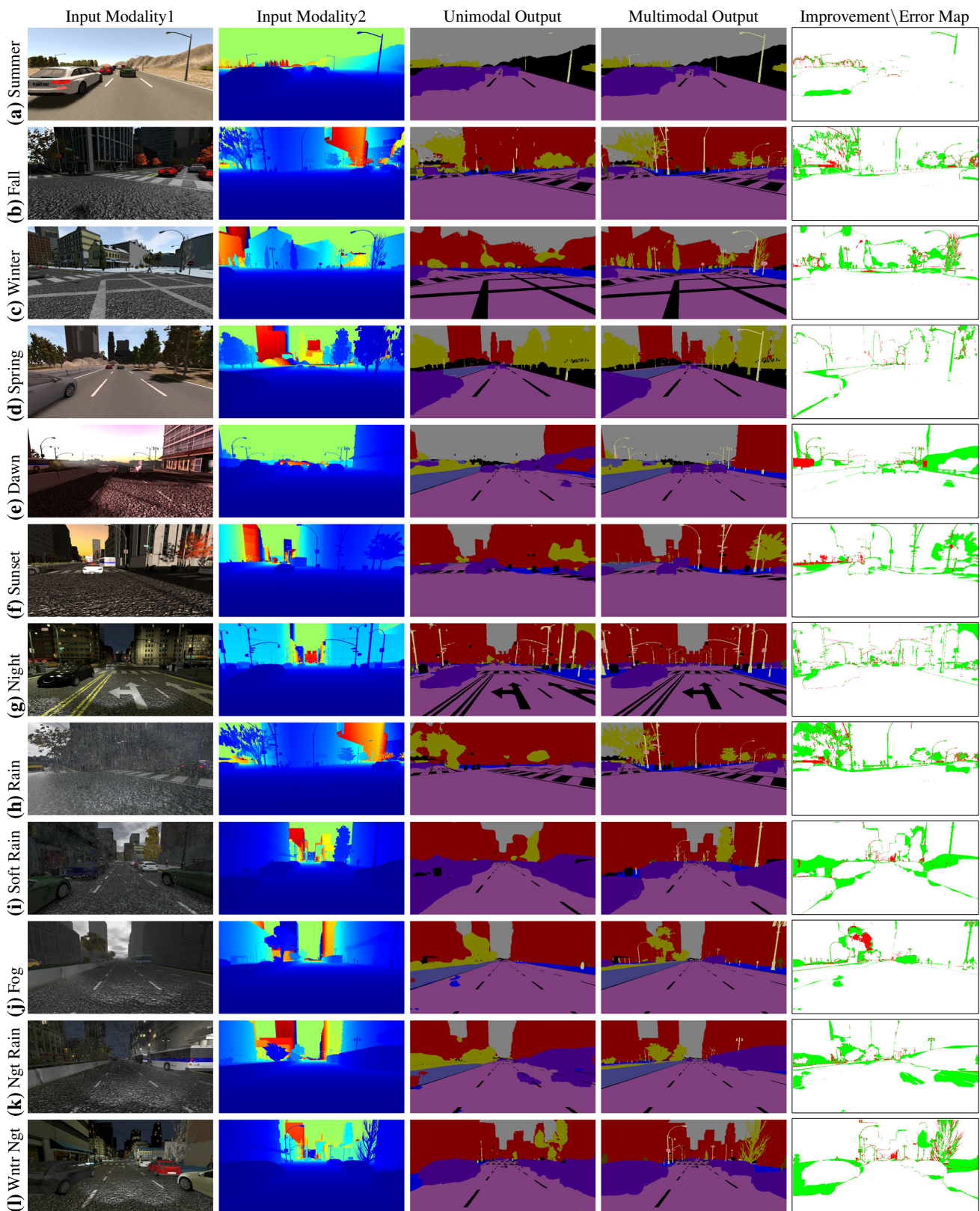
The final two rows in Fig. 21 show interesting failure modes where the multimodal fusion model demonstrates incorrect predictions. In Fig. 21k, we show an example from the Cityscapes dataset which contains an extremely thin *fence* connected by wires along the median of the road. The thin wires are not captured by the depth modality and it is visually infeasible to detect it from the RGB image. Moreover due to its thin structure, the vehicles on the opposite lane are clearly visible. This causes both the unimodal and multimodal models to partially segment the vehicles behind the fence which causes incorrect predictions according to the groundtruth.



**Fig. 21** Qualitative multimodal fusion results in comparison to the output of the unimodal visual RGB model on each of the five datasets that we benchmark on. The last two rows show failure modes. In addition to the segmentation output, we also show the improvement/error map which denotes the misclassified pixels in red and the pixels that are mis-

classified by the best performing state-of-the-art model but correctly predicted by AdapNet++ model in green. The legend for the segmented labels correspond to the colors shown in the benchmarking tables in Sect. 5.3 (Color figure online)





**Fig. 22** Qualitative multimodal semantic segmentation results in comparison the model trained on visual RGB images on the Synthia-Seasons dataset. In addition to the segmentation output, we also show the improvement/error map which denotes the misclassified pixels in red

and the pixels that are misclassified by the best performing state-of-the-art model but correctly predicted by AdapNet++ in green. The legend for the segmented labels correspond to the colors shown in the benchmarking tables in Sect. 5.3 (Color figure online)

However, we can see that the multimodal model still captures more of the *fence* structure than the unimodal model. In Fig. 211, we show an example from the SUN RGB-D dataset in which misclassifications are produced due to inconspicuous classes. The scene contains two object categories that have very similar appearance in some scenes, namely, *chair* and *sofa*. The *chair* class is denoted in dark green, while the *sofa* class is denoted in light green. As we see in this scene, a single-person *sofa* is considered to be a *chair* according to the groundtruth, whereas only the longer *sofa* in the middle is considered to be in the *sofa* class. In this scene, the single person *sofa* is adjacent to the longer *sofa* which causes the network to predict the pixels on both of them as the *sofa* class.

### 5.12 Visualizations Across Seasons and Weather Conditions

In this section, we present qualitative results on the Synthia-Sequences dataset that contains video sequences of 12 different seasons and weather conditions. We visualize the segmentation output of the multimodal and unimodal models for which the qualitative results are shown in Table 19. For this experiment, the models were trained on the Synthia-Rand-Cityscapes dataset and only evaluated on the Synthia-Sequences dataset. The Synthia-Sequences dataset contains a diverse set of conditions such as *summer, fall, winter, spring, dawn, sunset, night, rain, soft rain, fog, night rain* and *winter night*. We show qualitative evaluations on each of these conditions by comparing the multimodal segmentation performance with the output obtained from the unimodal visual RGB model in Fig. 22. This aim of this experiment is twofold: to study the robustness of the model to adverse perceptual conditions such as rain, snow, fog and nighttime; Secondly, to evaluate the generalization of the model to unseen scenarios.

From the examples shown in Fig. 22, we can see the diverse nature of the scenes containing environments such as highway driving, inner-city with skyscrapers and small sized cities. The visual RGB images in all of the scenes show the changing weather conditions that cause vegetation to change color in Fig. 22b, snow on the ground and leafless trees in Fig. 22c, glaring light due to sunrise in Fig. 22c, orange hue due to sunset in Fig. 22f, dark scene with isolated lights in Fig. 22g, noisy visibility due to rain in Fig. 22h and blurred visibility due to fog in Fig. 22j. Even for humans it is extremely hard to identify objects in some of these environments. The third column shows the output obtained from the unimodal visual RGB model. The output shows significant misclassifications in scenes that contain rain, fog, snow or nighttime, whereas the multimodal RGB-D model precisely segments the scene by leveraging the more stable depth features. The improvement map in green shown in the last column of Fig. 22, demonstrates substantial improvement over unimodal segmentation and minimal error for multi-

modal segmentation. The error is noticeable only along the boundaries of objects that are far away, which can be remedied using a higher resolution input image. Figure 22e, j show partial failure cases. In the first example in Fig. 22e, the occluded bus on the left is misclassified as a *fence* due to its location beyond the sidewalk, where often fences appear in the same configuration. While, in Fig. 22j, a segment of the *vegetation* several meters away is misclassified as a part of the *building* behind. However, overall the multimodal network is able to generalize to unseen environments and visibility conditions demonstrating the efficacy of our approach.

## 6 Conclusion

In this paper, we proposed an architecture for multimodal semantic segmentation that incorporates our self-supervised model adaptation blocks which dynamically adapt the fusion of features from modality-specific streams at various intermediate network stages in order to optimally exploit complementary features. Our fusion mechanism is simultaneously sensitive to critical factors that influence the fusion, including the object category, its spatial location and the environmental scene context in order to fuse only the relevant complementary information. We also introduced a channel attention mechanism for better correlating the fused mid-level modality-specific encoder features with the high-level decoder features for object boundary refinement. Moreover, as the fusion mechanism is self-supervised, we demonstrated that it effectively generalizes to the fusion of different modalities, beyond the commonly employed RGB-D data and across different environments ranging from urban driving scenarios to indoor scenes and unstructured forested environments.

In addition, we presented a computationally efficient unimodal semantic segmentation architecture that consists of an encoder with our multiscale residual units and an efficient atrous spatial pyramid module, complemented by a strong decoder with skip refinement stages. Our proposed multiscale residual units outperform the commonly employed multigrid method and our proposed efficient atrous spatial pyramid pooling achieves a  $10 \times$  reduction in the number of parameters with a simultaneous increase in performance compared to the standard atrous spatial pyramid pooling. Additionally, we proposed a holistic network-wide pruning approach to further compress our model to enable efficient deployment. We presented exhaustive theoretical analysis, visualizations, quantitative and qualitative results on Cityscapes, Synthia, SUN RGB-D, ScanNet and Freiburg Forest datasets. The results demonstrate that our unimodal AdapNet++ architecture achieves state-of-the-art performance on Synthia, ScanNet and Freiburg Forest benchmarks while demonstrating comparable performance on Cityscapes and SUN RGB-D benchmarks with a significantly lesser number of param-



ters and a substantially faster inference time in comparison to other state-of-the-art models. More importantly, our multimodal semantic segmentation architecture sets the new state-of-the-art on all the aforementioned benchmarks, while demonstrating exceptional robustness in adverse perceptual conditions.

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Anwar, S., Hwang, K., & Sung, W. (2017). Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3), 32.
- Audebert, N., Le Saux, B., & Lefèvre, S. (2018). Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140, 20–32.
- Badrinarayanan, V., Kendall, A., & Cipolla, R. (2015). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint [arXiv:1511.00561](https://arxiv.org/abs/1511.00561).
- Boniardi, F., Valada, A., Mohan, R., Caselitz, T., & Burgard, W. (2019). Robot localization in floor plans using a room layout edge extraction network. arXiv preprint [arXiv:1903.01804](https://arxiv.org/abs/1903.01804).
- Brostow, G. J., Shotton, J., Fauqueur, J., & Cipolla, R. (2008). Segmentation and recognition using structure from motion point clouds. In D. Forsyth, P. Torr, & A. Zisserman (Eds.), *Proceedings of the European conference on computer vision*.
- Bulò, S. R., Porzi, L., & Kotschieder, P. (2018). In-place activated batchnorm for memory-optimized training of dnns. In *Proceedings of the conference on computer vision and pattern recognition*.
- Chattopadhyay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2017). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. arXiv preprint [arXiv:1710.11063](https://arxiv.org/abs/1710.11063).
- Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint [arXiv:1606.00915](https://arxiv.org/abs/1606.00915).
- Chen, L. C., Collins, M., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., & Shlens, J. (2018a). Searching for efficient multi-scale architectures for dense image prediction. In *Advances in neural information processing systems* (pp. 8713–8724).
- Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017). Rethinking atrous convolution for semantic image segmentation. arXiv preprint [arXiv:1706.05587](https://arxiv.org/abs/1706.05587).
- Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018b). Encoder–decoder with atrous separable convolution for semantic image segmentation. arXiv preprint [arXiv:1802.02611](https://arxiv.org/abs/1802.02611).
- Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. arXiv preprint [arXiv:1610.02357](https://arxiv.org/abs/1610.02357).
- Cichy, R. M., Pantazis, D., & Oliva, A. (2016). Similarity-based fusion of meg and fmri reveals spatio-temporal dynamics in human cortex during visual object recognition. *Cerebral Cortex*, 26(8), 3563–3579.
- Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., et al. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the conference on computer vision and pattern recognition*.
- Coupric, C., Farabet, C., Najman, L., & LeCun, Y. (2013). Indoor semantic segmentation using depth information. arXiv preprint [arXiv:1301.3572](https://arxiv.org/abs/1301.3572).
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., & Nießner, M. (2017). Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the conference on computer vision and pattern recognition*.
- Dai, A., & Nießner, M. (2018). 3d-mv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. arXiv preprint [arXiv:1803.10409](https://arxiv.org/abs/1803.10409).
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proceedings of the conference on computer vision and pattern recognition*.
- Eitel, A., Springenberg, J. T., Spinello, L., Riedmiller, M. A., & Burgard, W. (2015). Multimodal deep learning for robust rgb-d object recognition. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems*.
- Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1), 98–136.
- Farabet, C., Coupric, C., Najman, L., & LeCun, Y. (2012). Scene parsing with multiscale feature learning, purity trees, and optimal covers. In *Proceedings of the international conference on machine learning*.
- Fei-Fei, L., Koch, C., Iyer, A., & Perona, P. (2004). What do we see when we glance at a scene? *Journal of Vision*, 4(8), 863–863.
- Fulkerson, B., Vedaldi, A., & Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the international conference on computer vision*.
- Ghiasi, G., & Fowlkes, C. C. (2016). Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European conference on computer vision* (pp. 519–534).
- Grangier, D., Bottou, L., & Collobert, R. (2009). Deep convolutional networks for scene parsing. In *ICML workshop on deep learning*.
- Gupta, S., Girshick, R., Arbeláez, P., & Malik, J. (2014). Learning rich features from rgb-d images for object detection and segmentation. In *Proceedings of the European conference on computer vision*.
- Hazirbas, C., Ma, L., Domokos, C., & Cremers, D. (2016). Fusernet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Proceedings of the Asian conference on computer vision*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015a). Deep residual learning for image recognition. In *Proceedings of the conference on computer vision and pattern recognition*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015b). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the international conference on computer vision*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015c). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904–1916.
- He, K., Zhang, X., Ren, S., Sun, J. (2016). Identity mappings in deep residual networks. In *Proceedings of the European conference on computer vision* (pp. 630–645).
- Hermans, A., Floros, G., & Leibe, B. (2014). Dense 3d semantic mapping of indoor scenes from rgb-d images. In *Proceedings of the IEEE international conference on robotics and automation*.
- Hu, J., Shen, L., & Sun, G. (2017). Squeeze-and-excitation networks. arXiv preprint [arXiv:1709.01507](https://arxiv.org/abs/1709.01507).
- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the conference on computer vision and pattern recognition* (pp. 7132–7141).
- Huete, A., Justice, C., & Van Leeuwen, W. (1999). Modis vegetation index (mod13). *Algorithm Theoretical Basis Document*, 3, 213.

- Janoch, A., Karayev, S., Jia, Y., Barron, J. T., Fritz, M., Saenko, K., et al. (2013). A category-level 3d object dataset: Putting the kinect to work. In *Proceedings of the IEEE international conference on consumer depth cameras for computer vision* (pp. 141–165).
- Kim, D. K., Maturana, D., Uenoyama, M., & Scherer, S. (2017). Season-invariant semantic segmentation with a deep multimodal network. In *Field and service robotics*.
- Kohli, P., Torr, P. H., et al. (2009). Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3), 302–324.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).
- Ku, J., Harakeh, A., & Waslander, S. L. (2018). In defense of classical image processing: Fast depth completion on the cpu. arXiv preprint [arXiv:1802.00036](https://arxiv.org/abs/1802.00036).
- LeCun, Y., Denker, J. S., & Solla, S. A. (1990). Optimal brain damage. In *Advances in neural information processing systems* (pp. 598–605).
- Lee, C. Y., Xie, S., Gallagher, P., Zhang, Z., & Tu, Z. (2015). Deeply-supervised nets. In *Artificial intelligence and statistics* (pp. 562–570).
- Li, H., Kadav, A., Durdanovic, I., Samet, H., & Graf, H. P. (2016). Pruning filters for efficient convnets. arXiv preprint [arXiv:1608.08710](https://arxiv.org/abs/1608.08710)
- Li, Z., Gan, Y., Liang, X., Yu, Y., Cheng, H., & Lin, L. (2016). Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *Proceedings of the European conference on computer vision*.
- Liang-Chieh, C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. (2015). Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International conference on learning representations*.
- Lin, G., Milan, A., Shen, C., & Reid, I. D. (2017). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the conference on computer vision and pattern recognition*.
- Lin, M., Chen, Q., & Yan, S. (2013). Network in network. arXiv preprint [arXiv:1312.4400](https://arxiv.org/abs/1312.4400).
- Liu, W., Rabinovich, A., & Berg, A. C. (2015). Parsenet: Looking wider to see better. arXiv preprint [arXiv:1506.04579](https://arxiv.org/abs/1506.04579).
- Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., & Zhang, C. (2017). Learning efficient convolutional networks through network slimming. In *Proceedings of the international conference on computer vision*.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the conference on computer vision and pattern recognition*.
- Molchanov, P., Tyree, S., Karras, T., Aila, T., & Kautz, J. (2017). Pruning convolutional neural networks for resource efficient inference. *Proceedings of the international conference on learning representation*.
- Munoz, D., Bagnell, J. A., & Hebert, M. (2012). Co-inference for multimodal scene analysis. In *Proceedings of the European conference on computer vision*.
- Noh, H., Araujo, A., Sim, J., Weyand, T., & Han, B. (2017). Largescale image retrieval with attentive deep local features. In *Proceedings of the IEEE International conference on computer vision* (pp. 3456–3465).
- Noh, H., Hong, S., & Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the international conference on computer vision* (pp. 1520–1528).
- Oliveira, G., Valada, A., Bollen, C., Burgard, W., & Brox, T. (2016). Deep learning for human part discovery in images. In *Proceedings of the IEEE international conference on robotics and automation*.
- Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint [arXiv:1606.02147](https://arxiv.org/abs/1606.02147).
- Pinheiro, P. O., & Collobert, R. (2014). Recurrent convolutional neural networks for scene labeling. In *Proceedings of the international conference on machine learning*.
- Plath, N., Toussaint, M., & Nakajima, S. (2009). Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the international conference on machine learning*.
- Qi, X., Liao, R., Jia, J., Fidler, S., & Urtasun, R. (2017). 3d graph neural networks for rgbd semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5199–5208).
- Radwan, N., Valada, A., & Burgard, W. (2018a). Multimodal interaction-aware motion prediction for autonomous street crossing. arXiv preprint [arXiv:1808.06887](https://arxiv.org/abs/1808.06887).
- Radwan, N., Valada, A., & Burgard, W. (2018b). Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters (RA-L)*, 3(4), 4407–4414.
- Ren, X., Bo, L., & Fox, D. (2012). Rgb-d scene labeling: Features and algorithms. In *Proceedings of the conference on computer vision and pattern recognition*.
- Romera, E., Alvarez, J. M., Bergasa, L. M., & Arroyo, R. (2018). Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1), 263–272.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241).
- Ros, G., Sellart, L., Materzynska, J., Vazquez, D., & Lopez, A. M. (2016). The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the conference on computer vision and pattern recognition*.
- Running, S. W., Nemani, R., Glassy, J. M., & Thornton, P. E. (1999). *Modis daily photosynthesis (psn) and annual net primary production (npp) product (mod17) algorithm theoretical basis document*. University of Montana, SCF At-Launch Algorithm ATBD Documents.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the conference on computer vision and pattern recognition* (pp. 4510–4520).
- Schneider, L., Jasch, M., Fröhlich, B., Weber, T., Franke, U., Pollefeys, M., et al. (2017). Multimodal neural networks: Rgb-d for semantic segmentation and object detection. In P. Sharma & F. M. Bianchi (Eds.), *Image analysis* (pp. 98–109). Cham: Springer.
- Shotton, J., Johnson, M., & Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *Proceedings of the conference on computer vision and pattern recognition*.
- Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *Proceedings of the European conference on computer vision*.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- Song, S., Lichtenberg, S. P., & Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the conference on computer vision and pattern recognition* (Vol. 5, p. 6).
- Sturgess, P., Alahari, K., Ladicky, L., & Torr, P. H. S. (2009). Combining appearance and structure from motion features for road scene understanding. In *Proceedings of the British machine vision conference*.
- Valada, A., Dhall, A., & Burgard, W. (2016a). Convolutional mixture of deep experts for robust semantic segmentation. In *IEEE/RSJ International conference on intelligent robots and systems (IROS) workshop, state estimation and terrain perception for all terrain mobile robots*.

- Valada, A., Oliveira, G., Brox, T., & Burgard, W. (2016b). Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *Proceedings of the international symposium for experimental robotics*.
- Valada, A., Oliveira, G., Brox, T., & Burgard, W. (2016c). Towards robust semantic segmentation using deep fusion. In *Robotics: Science and systems (RSS 2016) workshop, are the sceptics right? Limits and potentials of deep learning in robotics*.
- Valada, A., Vertens, J., Dhall, A., & Burgard, W. (2017). Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *Proceedings of the IEEE international conference on robotics and automation*.
- Wen, W., Wu, C., Wang, Y., Chen, Y., & Li, H. (2016). Learning structured sparsity in deep neural networks. In *Advances in neural information processing systems* (pp. 2074–2082).
- Xiang, Y., & Fox, D. (2017). Da-rnn: Semantic mapping with data associated recurrent neural networks. arXiv preprint [arXiv:1703.03098](https://arxiv.org/abs/1703.03098).
- Xiao, J., Owens, A., & Torralba, A. (2013). Sun3d: A database of big spaces reconstructed using sfm and object labels. In *Proceedings of the international conference on computer vision*.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the conference on computer vision and pattern recognition* (pp. 5987–5995).
- Yang, M., Yu, K., Zhang, C., Li, Z., & Yang, K. (2018). Denseaspp for semantic segmentation in street scenes. In *Proceedings of the conference on computer vision and pattern recognition* (pp. 3684–3692).
- Yu, F., & Koltun, V. (2016). Multi-scale context aggregation by dilated convolutions. In *International conference on learning representations*.
- Zhang, C., Wang, L., & Yang, R. (2010). Semantic segmentation of urban scenes using dense depth maps. In K. Daniilidis, P. Maragos, & N. Paragios (Eds.), *Proceedings of the European conference on computer vision*.
- Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the conference on computer vision and pattern recognition*.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2014). Object detectors emerge in deep scene cnns. arXiv preprint [arXiv:1412.6856](https://arxiv.org/abs/1412.6856).
- Zhuang, Y., Yang, F., Tao, L., Ma, C., Zhang, Z., Li, Y., et al. (2018). Dense relation network: Learning consistent and context-aware representation for semantic image segmentation. In *2018 25th IEEE international conference on image processing (ICIP)* (pp. 3698–3702).

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.