FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-based CNN Architecture

Caner Hazirbas†, Lingni Ma†, Csaba Domokos, and Daniel Cremers

Technical University of Munich, Germany
{hazirbas,lingni,domokos,cremers}@cs.tum.edu

Abstract. In this paper we address the problem of semantic labeling of indoor scenes on RGB-D data. With the availability of RGB-D cameras, it is expected that additional depth measurement will improve the accuracy. Here we investigate a solution how to incorporate complementary depth information into a semantic segmentation framework by making use of convolutional neural networks (CNNs). Recently encoder-decoder type fully convolutional CNN architectures have achieved a great success in the field of semantic segmentation. Motivated by this observation we propose an encoder-decoder type network, where the encoder part is composed of two branches of networks that simultaneously extract features from RGB and depth images and fuse depth features into the RGB feature maps as the network goes deeper. Comprehensive experimental evaluations demonstrate that the proposed fusion-based architecture achieves competitive results with the state-of-the-art methods on the challenging SUN RGB-D benchmark obtaining 76.27% global accuracy, 48.30% average class accuracy and 37.29% average intersection-over-union score.

1 Introduction

Visual scene understanding in a glance is one of the most amazing capability of the human brain. In order to model this ability, semantic segmentation aims at giving a class label for each pixel on the image according to its semantic meaning. This problem is one of the most challenging tasks in computer vision, and has received a lot of attention from the computer vision community [1,2,3,4,5,6,7].

Convolutional neural networks (CNNs) have recently attained a breakthrough in various classification tasks such as semantic segmentation. CNNs have been shown to be powerful visual models that yields hierarchies of features. The key success of this model mainly lies in its general modeling ability for complex visual scenes. Currently CNN-based approaches [3,8,4] provide the state-of-the-art performance in several semantic segmentation benchmarks. In contrast to CNN models, by applying hand-crafted features one can generally achieve rather limited accuracy.

Utilizing depth additional to the appearance information (*i.e.* RGB) could potentially improve the performance of semantic segmentation, since the depth

† These authors contributed equally.

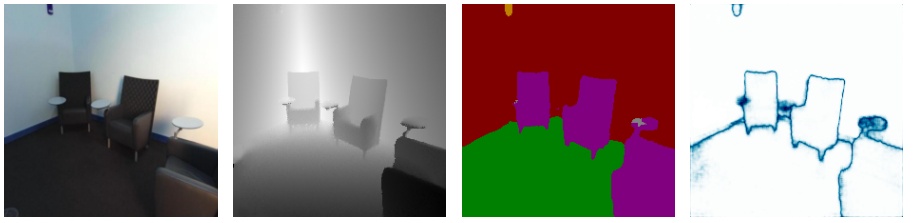


Fig. 1: An exemplar output of FuseNet. From left to right: input RGB and depth images, the predicted semantic labeling and the probability of the corresponding labels, where white and blue denote high and low probability, respectively.

channel has complementary information to RGB channels, and encodes structural information of the scene. The depth channel can be easily captured with low cost RGB-D sensors. In general object classes can be recognized based on their color and texture attributes. However, the auxiliary depth may reduce the uncertainty of the segmentation of objects having similar appearance information. Couprie *et al*. [9] observed that the segmentation of classes having similar depth, appearance and location is improved by making use of the depth information too, but it is better to use only RGB information to recognize object classes containing high variability of their depth values. Therefore, the optimal way to fuse RGB and depth information has been left an open question.

In this paper we address the problem of indoor scene understanding assuming that both RGB and depth information simultaneously available (see Figure 1). This problem is rather crucial in many perceptual applications including robotics. We remark that although indoor scenes have rich semantic information, they are generally more challenging than outdoor scenes due to more severe occlusions of objects and cluttered background. For example, indoor object classes, such as *chair*, *dining table* and *curtain* are much harder to recognize than outdoor classes, such as *car*, *road*, *building* and *sky*.

The contribution of the paper can be summarized as follows:

- We investigate a solution how to incorporate complementary depth information into a semantic segmentation framework. For this sake we propose an encoder-decoder type network, referred to as FuseNet, where the encoder part is composed of two branches of networks that simultaneously extract features from RGB and depth images and fuse depth features into the RGB feature maps as the network goes deeper (see Figure 2).
- We propose and examine two different ways for fusion of the RGB and depth channels. We also analyze the proposed network architectures, referred to as dense and sparse fusion (see Figure 3), in terms of the level of fusion.
- We experimentally show that our proposed method is successfully able to fuse RGB and depth information for semantic segmentation also on cluttered indoor scenes. Moreover, our method achieves competitive results with state-of-the-art methods in terms of segmentation accuracy evaluated on the challenging SUN RGB-D dataset [10].

2 Related Work

A fully convolutional network (FCN) architecture has been introduced in [3] that combines semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations by applying end-to-end training. Noh *et al.* [6] have proposed a novel network architecture for semantic segmentation, referred to as DeconvNet, which alleviates the limitations of fully convolutional models (*e.g.*, very limited resolution of labeling). DeconvNet is composed of deconvolution and unpooling layers on top of the VGG 16-layer net [11]. To retrieve semantic labeling on the full image size, Zeiler *et al.* [12] have introduced a network composed of deconvolution and unpooling layers. Concurrently, a very similar network architecture has been presented [13] based on the VGG 16-layer net [11], referred to as *SegNet*. In contrast to DeconvNet, SegNet consists of smoothed unpooled feature maps with convolution instead of deconvolution. Kendall *et al.* [14] further improved the segmentation accuracy of SegNet by applying dropout [15] during test time [16].

Some recent semantic segmentation algorithms combine the strengths of CNN and conditional random field (CRF) models. It has been shown that the poor pixel classification accuracy, due to the invariance properties that make CNNs good for high level tasks, can be overcome by combining the responses of the CNN at the final layer with a fully connected CRF model [8]. CNN and CRF models have also been combined in [4]. More precisely, the method proposed in [4] applies mean field approximation as the inference for a CRF model with Gaussian pairwise potentials, where the mean field approximation is modeled as a recurrent neural network, and the defined network is trained end-to-end refining the weights of the CNN model. Recently, Lin *et al.* [7] have also combined CNN and CRF models for learning patch-patch context between image regions, and have achieved the current state-of-the-art performance in semantic segmentation. One of the main ideas in [7] is to define CNN-based pairwise potential functions to capture semantic correlations between neighboring patches. Moreover, efficient piecewise training is applied for the CRF model in order to avoid repeated expensive CRF inference during the course of back-propagation.

In [2] a feed-forward neural network has been proposed for scene labeling. The long range (pixel) label dependencies can be taken into account by capturing sufficiently large input context patch, around each pixel to be labeled. The method [2] relies on a recurrent convolutional neural networks (RCNN), *i.e.* a sequential series of networks sharing the same set of parameters. Each instance takes as input both an RGB image and the predictions of the previous instance of the network. RCNN-based approaches are known to be difficult to train, in particular, with large data, since long-term dependencies are vanished while the information is accumulated by the recurrence [5].

Byeon *et al.* [5] have presented long short term memory (LSTM) recurrent neural networks for natural scene images taking into account the complex spatial dependencies of labels. LSTM networks have been commonly used for sequence classification. These networks include recurrently connected layers to learn the dependencies between two frames, and then transfer the probabilistic inference

to the next frame. This allows to easily memorize the context information for long periods of time in sequence data. It has been shown [5] that LSTM networks can be generalized well to any vision-based task and efficiently capture local and global contextual information with a low computational complexity.

State-of-the-art CNNs have the ability to perform segmentation on different kinds of input sources such as RGB or even RGB-D. Therefore a trivial way to incorporate depth information would be to stack it to the RGB channels and train the network on RGB-D data assuming a four-channel input. However, it would not fully exploit the structure of the scene encoded by the depth channel. This will be also shown experimentally in Section 4. By making use of deeper and wider network architecture one can expect the increase of the robustness and the accuracy. Hence, one may define a network architecture with more layers. Nevertheless, this approach would require huge dataset in order to learn all the parameter making the training infeasible even in the case when the parameters are initialized with a pre-trained network.

2.1 The State of the Arts on RGB-D Data

A new representation of the depth information has been presented by Gupta *et al.* [1]. This representation, referred to as HHA, consists of three channels: disparity, height of the pixels and the angle between of normals and the gravity vector based on the estimated ground floor, respectively. By making use of the HHA representation, a superficial improvement was achieved in terms of segmentation accuracy [1]. On the other hand, the information retrieved only from the RGB channels still dominates the HHA representation. As we shall see in Section 4, the HHA representation does not hold more information than the depth itself. Furthermore, computing HHA representation requires high computational cost. In this paper we investigate a better way of exploiting depth information with less computational burden.

Li *et al.* [17] have introduced a novel LSTM Fusion (LSTM-F) model that captures and fuses contextual information from photometric and depth channels by stacking several convolutional layers and an LSTM layer. The memory layer encodes both short- and long-range spatial dependencies in an image along vertical direction. Moreover, another LSTM-F layer integrates the contexts from different channels and performs bi-directional propagation of the fused vertical contexts. In general, these kinds of architectures are rather complicated and hence more difficult to train. In contrast to recurrent networks, we propose a simpler network architecture.

3 FuseNet: Unified CNN Framework for Fusing RGB and Depth Channels

We aim to solve the semantic segmentation problem on RGB-D images. We define the label set as $\mathcal{L} = \{1, 2, \ldots, K\}$. We assume that we are given a training set $\{(\mathbf{X}_i, \mathbf{Y}_i) \mid \mathbf{X}_i \in \mathbb{R}^{H \times W \times 4}, \mathbf{Y}_i \in \mathcal{L}^{H \times W} \text{ for all } i = 1, \ldots, M\}$ consisting of

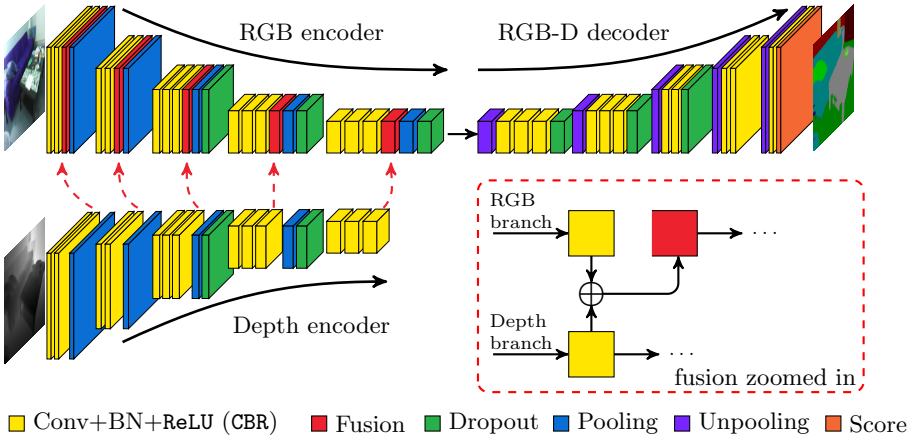


Fig. 2: The architecture of the proposed FuseNet. Colors indicate the layer type. The network contains two branches to extract features from RGB and depth images, and the feature maps from depth is constantly fused into the RGB branch, denoted with the red arrows. In our architecture, the fusion layer is implemented as an element-wise summation, demonstrated in the dashed box.

M four-channel RGB-D images ($\mathbf{X}_i$), having the same size $H \times W$, along with the ground-truth labeling ($\mathbf{Y}_i$). Moreover, we assume that the pixels are drawn as *i.i.d.* samples following a categorical distribution. Based on this assumption, we may define a CNN model to perform multinomial logistic regression.

The network extracts features from the input layer and through filtering provides classification score for each label as an output at each pixel. We model the network as a composition of functions corresponding to L layers with parameters denoted by $\mathbf{W} = [\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(L)}]$, that is

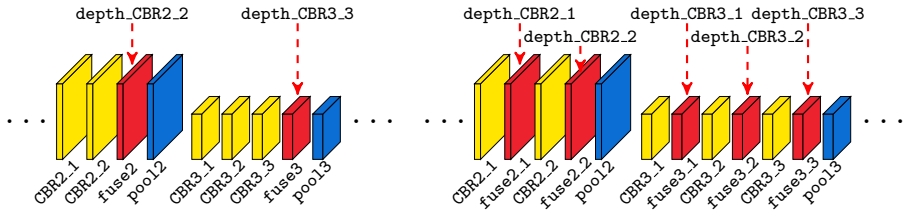
$$f(\mathbf{x}; \mathbf{W}) = g^{(L)}(g^{(L-1)}(\cdots g^{(2)}(g^{(1)}(\mathbf{x}; \mathbf{w}^{(1)}); \mathbf{w}^{(2)}) \cdots ; \mathbf{w}^{(L-1)}); \mathbf{w}^{(L)}) \,. \quad (1)$$

The classification score of a pixel $\mathbf{x}$ for a given class c is obtained from the function $f_c(\mathbf{x}; \mathbf{W})$, which is the cth component of $f(\mathbf{x}; \mathbf{W})$. Using the *softmax* function, we can map this score to a probability distribution

$$p(c \mid \mathbf{x}, \mathbf{W}) = \frac{\exp\left(f_c(\mathbf{x}; \mathbf{W})\right)}{\sum_{k=1}^{K} \exp\left(f_k(\mathbf{x}; \boldsymbol{W})\right)} \,. \quad (2)$$

For the training of the network, *i.e.* learning the optimal parameters $\mathbf{W}^*$, the cross-entropy loss is used, which minimizes the KL-divergence between the predicted and the true class distribution:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{W}\|^2 - \frac{\lambda}{MHW} \sum_{i=1}^{M} \sum_{j=1}^{HW} \log p(y_{ij} \mid \mathbf{x}_{ij}, \mathbf{W}) \,,$$



(a) Sparse fusion (SF) (b) Dense fusion (DF)

Fig. 3: Illustration of different fusion strategies at the second (CBR2) and third (CBR3) convolution blocks of VGG 16-layer net. (a) The fusion layer is only inserted before each pooling layer. (b) The fusion layer is inserted after each CBR block.

where $\mathbf{x}_{ij} \in \mathbb{R}^4$ stands for the jth pixel of the ith training image and $y_{ij} \in \mathcal{L}$ is its ground-truth label. The hyper-parameter $\lambda > 0$ is chosen to apply weighting for the regularization of the parameters (*i.e.* L_2-norm of $\mathbf{W}$).

At inference, a probability distribution is predicted for each pixel via softmax normalization, defined in (2), and the labeling is calculated based on the highest class probability.

3.1 FuseNet Architecture

We propose an encoder-decoder type network architecture as shown in Figure 2. The proposed network has two major parts: 1) the *encoder* part extracts features and 2) the *decoder* part upsamples the feature maps back to the original input resolution. This encoder-decoder style has been already introduced in several previous works such as DeconvNet [6] and SegNet [13] and has achieved good segmentation performance. Although our proposed network is based on this type of architecture, we further consider to have two encoder branches. These two branches extract features from RGB and depth images. We note that the depth image is normalized to have the same value range as color images, *i.e.* into the interval of [0,255]. In order to combine information from both input modules, we fuse the feature maps from the depth branch into the feature maps of the RGB branch. We refer to this architecture as *FuseNet* (see Figure 2).

The encoder part of FuseNet resembles the 16-layer VGG net [11], except of the fully connected layers `fc6`, `fc7` and `fc8`, since the fully connected layers reduce the resolution with a factor of 49, which increases the difficulty of the upsampling part. In our network, we always use batch normalization (`BN`) after convolution (`Conv`) and before rectified linear unit[1] (`ReLU`) to reduce the internal covariate shift [18]. We refer to the combination of convolution, batch normalization and `ReLU` as `CBR` block, respectively. The `BN` layer first normalizes the feature maps to have zero-mean and unit-variance, and then scales and shifts

[1] The rectified linear unit is defined as $\sigma(x) = \max(0, x)$.

them afterwards. In particular, the scale and shift parameters are learned during training. As a result, color features are not overwritten by depth features, but the network learns how to combine them in an optimal way.

The decoder part is a counterpart of the encoder part, where memorized unpooling is applied to upsample the feature maps. In the decoder part, we again use the `CBR` blocks. We also did experiments with deconvolution instead of convolution, and observed very similar performance. As proposed in [14], we also apply dropout in both the encoder and the decoder parts to further boost the performance. However, we do not use dropout during test time.

The key ingredient of the FuseNet architecture is the fusion block, which combines the feature maps of the depth branch and the RGB branch. The fusion layer is implemented as element-wise summation. In FuseNet, we always insert the fusion layer after the `CBR` block. By making use of fusion the discontinuities of the features maps computed on the depth image are added into the RGB branch in order to enhance the RGB feature maps. As it can be observed in many cases, the features in the color domain and in the geometric domain complement each other. Based on this observation, we propose two fusion strategies: a) dense fusion (`DF`), where the fusion layer is added after each `CBR` block of the RGB branch. b) sparse fusion (`SF`), where the fusion layer is only inserted before each pooling. These two strategies are illustrated in Figure 3.

3.2 Fusion of Feature Maps

In this section, we reason the fusion of the feature maps between the RGB and the depth branches. To utilize depth information a simple way would be just stacking the RGB and depth images into a four-channel input. However, we argue that by fusing RGB and depth information the feature maps are usually more discriminant than the ones obtained from the stacked input.

As we introduced before in Equation (1), each layer is modeled as a function g that maps a set of input $\mathbf{x}$ to a set of output $\mathbf{a}$ with parameter $\mathbf{w}$. We denote the kth feature map in the lth layer by $g_k^{(l)}$. Suppose that the given layer operation consists of convolution and `ReLU`, therefore

$$\mathbf{x}_k^{(l+1)} = g_k^{(l)}(\mathbf{x}^{(l)}; \mathbf{w}_k^{(l)}) = \sigma(\langle \mathbf{w}_k^{(l)}, \mathbf{x}^{(l)} \rangle + b_k^{(l)}) \, .$$

If the input is a four-channel RGB-D image, then the feature maps can be decomposed as $\mathbf{x} = [\boldsymbol{a}^\intercal \ \boldsymbol{b}^\intercal]^\intercal$, where $\boldsymbol{a} \in \mathbb{R}^{d_1}$, $\boldsymbol{b} \in \mathbb{R}^{d_2}$ with $d_1 + d_2 = d := \dim(\mathbf{x})$ are features learned from the color channels and from the depth channel, respectively. According to this observation, we may write that

$$\begin{aligned}
\mathbf{x}_k^{(l+1)} &= \sigma(\langle \mathbf{w}_k^{(l)}, \mathbf{x}^{(l)} \rangle + b_k^{(l)}) = \sigma(\langle \mathbf{u}_k^{(l)}, \mathbf{a}^{(l)} \rangle + c_k^{(l)} + \langle \mathbf{v}_k^{(l)}, \mathbf{b}^{(l)} \rangle + d_k^{(l)}) \\
&= \max\left(\mathbf{0}, \langle \mathbf{u}_k^{(l)}, \mathbf{a}^{(l)} \rangle + c_k^{(l)} + \langle \mathbf{v}_k^{(l)}, \mathbf{b}^{(l)} \rangle + d_k^{(l)})\right) \\
&\leq \max(\mathbf{0}, \langle \mathbf{u}_k^{(l)}, \mathbf{a}^{(l)} \rangle + c_k^{(l)}) + \max(\mathbf{0}, \langle \mathbf{v}_k^{(l)}, \mathbf{b}^{(l)} \rangle + d_k^{(l)}) \\
&= \sigma(\langle \mathbf{u}_k^{(l)}, \mathbf{a}^{(l)} \rangle + c_k^{(l)}) + \sigma(\langle \mathbf{v}_k^{(l)}, \mathbf{b}^{(l)} \rangle + d_k^{(l)}) \, ,
\end{aligned} \tag{3}$$

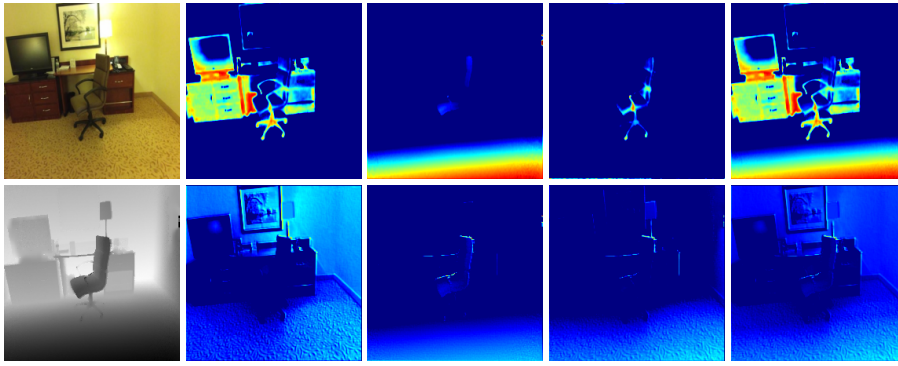


Fig. 4: Comparison of two out of 64 feature maps produced at the CBR1_1 layer. The features from RGB and depth mostly compensate each other, where the textureless region usually have rich structure features and structureless regions usually present texture features. This visually illustrates that the proposed fusion strategy better preserves the informative features from color and depth than applying element-wise summation followed by ReLU.

where we applied the decomposition of $\mathbf{w}_k^{(l)} = [\mathbf{u}_k^{(l)^\intercal} \ \mathbf{v}_k^{(l)^\intercal}]^\intercal$ and $b_k^{(l)} = c_k^{(l)} + d_k^{(l)}$.

Based on the inequality in (3), we show that the fusion of activations of the color and the depth branches (*i.e.* their element-wise summation) produces a stronger signal than the activation on the fused features. Nevertheless, the stronger activation does not necessarily lead to a better accuracy. However, with fusion, we do not only increase the neuron-wise activation values, but also preserve activations at different neuron locations. The intuition behind this can be seen by considering low-level features (*e.g.*, edges). Namely, due to the fact that the edges extracted in RGB and depth images are usually complementary to each other. One may combine the edges from both inputs to obtain more information. Consequently, these low-level features help the network to extract better high-level features, and thus enhance the ultimate accuracy.

To demonstrate the advantage of the proposed fusion, we visualize the feature maps produced by CBR1_1 in Figure 4, which corresponds to low-level feature extraction (*e.g.*, edges). As it can be seen the low-level features in RGB and depth are usually complementary to each other. For example, the textureless region can be distinguished by its structure, such as the lap against the wall, whereas the structureless region can be distinguished by the color, such as the painting on the wall. While combining the feature maps before the ReLU layer fail to preserve activations, however, the proposed fusion strategy, applied after the ReLU layer, preserves well all the useful information from both branches. Since low-level features help the network to extract better high-level ones, the proposed fusion thus enhances the ultimate accuracy.

4 Experimental Evaluation

In this section, we evaluate the proposed network through extensive experiments. For this purpose, we use the publicly available SUN RGB-D scene understanding benchmark [10]. This dataset contains 10335 synchronized RGB-D pairs, where pixel-wise annotation is available. The standard trainval-test split consists of 5050 images for testing and 5285 images for training/validation. This benchmark is a collection of images captured with different types of RGB-D cameras. The dataset also contains in-painted depth images, obtained by making use of multi-view fusion technique. In the experiments we used the standard training and test split with in-painted depth images. However, we excluded 587 training images that are originally obtained with RealSense RGB-D camera. This is due to the fact that raw depth images from the aforementioned camera consist of many invalid measurements, therefore in-painted depth images have many false values. We remark that the SUN RGB-D dataset is highly unbalanced in terms of class instances, where 16 out of 37 classes rarely present. To prevent the network from over-fitting towards unbalanced class distribution, we weighted the loss for each class with the median frequency class balancing according to [19]. In particular, the class *floormat* and *shower-curtain* have the least frequencies and they are the most challenging ones in the segmentation task. Moreover, approximately 0.25% pixels are not annotated and do not belong to any of the 37 target classes.

Training We trained the all networks end-to-end. Therefore images were resized to the resolution of 224×224. To this end we applied bilinear interpolation on the RGB images and nearest-neighbor interpolation on the depth images and the ground-truth labeling. The networks were implemented with the Caffe framework [20] and were trained with stochastic gradient descent (SGD) solver [21] using a batch size of 4. The input data was randomly shuffled after each epoch. The learning rate was initialized to 0.001 and was multiplied by 0.9 in every 50,000 iterations. We used a momentum of 0.9 and set weight decay to 0.0005. We trained the networks until convergence, when no further decrease in the loss was observed. The parameters in the encoder part of the network were fine-tuned from the VGG 16-layer model [11] pre-trained on the ImageNet dataset [22]. The original VGGNet requires a three-channel color image. Therefore, for different input dimensions we processed the weights of first layer (*i.e.* `conv1_1`) as follows:

i) averaged the weights along the channel for a single-channel depth input;
ii) stacked the weights with their average for a four-channel RGB-D input;
iii) duplicated the weights for a six-channel RGB-HHA input.

Testing We evaluated the results on the original 5050 test images. For quantitative evaluation, we used three criteria. Let TP, FP, FN denote the total number of true positive, false positive, false negative, respectively, and N denotes the total number of annotated pixels. We define the following three criteria:

Table 1: Segmentation results on the SUN RGB-D benchmark [10] in comparison to the state of the art. Our methods `DF1` and `SF5` outperforms most of the existing methods, except of the Context-CRF [7].

	Global	Mean	IoU
FCN-32s [3]	68.35	41.13	29.00
FCN-16s [3]	67.51	38.65	27.15
Bayesian SegNet [14] (RGB)	71.2	45.9	30.7
LSTM [17]	-	48.1	-
Context-CRF [7] (RGB)	78.4	53.4	42.3
FuseNet-`SF5`	76.27	48.30	37.29
FuseNet-`DF1`	73.37	50.07	34.02

i) Global accuracy, referred to as *global*, is the percentage of the correctly classified pixels, defined as

$$\text{Global} = \frac{1}{N} \sum_c \text{TP}_c \ , \quad c \in \{1...K\} \ .$$

ii) Mean accuracy, referred to as *mean*, is the average of classwise accuracy, defined as

$$\text{Mean} = \frac{1}{K} \sum_c \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \ .$$

iii) Intersection-over-union (IoU) is average value of the intersection of the prediction and ground truth regions over the union of them, defined as

$$\text{IoU} = \frac{1}{K} \sum_c \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c} \ .$$

Among these three measures, the global accuracy is relatively less informative due to the unbalanced class distribution. In general, the frequent classes receive a high score and hence dominate the less frequent ones. Therefore we also measured the average class accuracy and IoU score to provide a better evaluation of our method.

4.1 Quantitative Results

In the first experiment, we compared our FuseNet to the state-of-the-art methods. The results are presented in Table 1. We denote the SparseFusion and DenseFusion by `SF`, `DF`, respectively, following by the number of fusion layers used in the network (*e.g.*, `SF5`). The results shows that our FuseNet outperforms most of the existing methods with a significant margin. FuseNet is not as competitive in comparison to the Context-CRF [7]. However, it is also worth noting that the Context-CRF trains the network with a different loss function that corresponds to piecewise CRF training. It also requires mean-field approximation at

Table 2: Segmentation results of FuseNet in comparison to the networks trained with RGB, depth, HHA and their combinations. The second part of the table provides the results of variations of FuseNet. We show that FuseNet obtained significant improvements by extracting more informative features from depth.

Input	Global	Mean	IoU
Depth	69.06	42.80	28.49
HHA	69.21	43.23	28.88
RGB	72.14	47.14	32.47
RGB-D	71.39	49.00	31.95
RGB-HHA	73.90	45.57	33.64
FusetNet-SF1	75.48	46.15	35.99
FusetNet-SF2	75.82	46.44	36.11
FusetNet-SF3	76.18	47.10	36.63
FusetNet-SF4	76.56	48.46	37.76
FusetNet-SF5	76.27	48.30	37.29
FusetNet-DF1	73.37	50.07	34.02
FusetNet-DF2	73.31	49.39	33.97
FusetNet-DF3	73.37	49.46	33.52
FusetNet-DF4	72.83	49.53	33.46
FusetNet-DF5	72.56	49.86	33.04

the inference stage, followed by a dense fully connected CRF refinement to produce the final prediction. Applying the similar loss function and post-processing, FuseNet is likely to produce on-par or better results.

In the second experiment, we compare the FuseNet to network trained with different representation of depth, in order to further evaluate the effectiveness of depth fusion and different fusion variations. The results are presented in Table 2. It can be seen that stacking depth and HHA into color gives slight improvements over network trained with only color, depth or HHA. In contrast, with the depth fusion of FuseNet, we improve over a significant margin, in particular with respect to the IoU scores. We remark that the depth fusion is in particular useful as a replacement for HHA. Instead of preprocessing a single channel depth images to obtain hand crafted three-channel HHA representation, FuseNet learns high dimensional features from depth end-to-end, which is more informative as shown by experiments.

In Table 2, we also analyzed the performance of different variations of FuseNet. Since the original VGG 16-layer network has 5 levels of pooling, we increase the number of fusion layers as the network gets deeper. The experiments show that segmentation accuracy gets improved from `SF1` to `SF5`, however the increase appears saturated up to the fusion after the 4th pooling, *i.e.*, `SF4`. The possible reason behind the accuracy saturation is that depth already provides very distinguished features at low-level to compensate textureless regions in RGB, and we consistently fuse features extracted from depth into the RGB-branch. The same trend can be observed with `DF`.

In the third experiment, we further compare FuseNet-`SF5`, FuseNet-`DF1` to the network trained with RGB-D input. In Table 3 and 4, we report the class-wise accuracy and IoU scores of 37 classes, respectively. For class accuracy, all

Table 3: Classwise segmentation accuracy of 37 classes. We compare FuseNet-SF5, FuseNet-DF1 to the network trained with stacked RGB-D input.

	wall	floor	cabin	bed	chair	sofa	table	door	wdw	bslf	pic	cnter	blinds
RGB-D	77.19	93.90	**62.51**	74.62	71.22	59.09	**66.76**	42.27	62.73	29.51	64.66	**48.19**	**48.80**
SF5	**90.20**	**94.91**	61.81	**77.10**	**78.62**	**66.49**	65.44	**46.51**	62.44	**34.94**	67.39	40.37	43.48
DF1	82.39	93.88	56.97	73.76	78.02	62.85	60.60	45.43	**67.22**	28.79	**67.50**	39.89	44.73
	desk	shelf	ctn	drssr	pillow	mirror	mat	clthes	ceil	books	fridge	tv	paper
RGB-D	12.12	9.27	63.26	40.44	52.02	52.99	0.00	**38.38**	84.06	**57.05**	**34.90**	45.77	**41.54**
SF5	**25.63**	**20.28**	**65.94**	44.03	54.28	52.47	0.00	25.89	84.77	45.23	34.52	34.83	24.08
DF1	20.98	14.46	61.43	**48.63**	**58.59**	**55.96**	0.00	30.52	**86.23**	53.86	32.31	**53.13**	36.67
	towel	shwr	box	board	person	stand	toilet	sink	lamp	btub	bag	mean	
RGB-D	**27.92**	4.99	**31.24**	69.08	16.97	42.70	76.80	**69.41**	50.28	65.41	24.90	49.00	
SF5	21.05	**8.82**	21.94	57.45	19.06	37.15	76.77	68.11	49.31	73.23	12.62	48.30	
DF1	27.14	1.96	26.61	**66.36**	**30.91**	**43.89**	**81.38**	66.47	**52.64**	**74.73**	**25.80**	**50.07**	

Table 4: Classwise IoU scores of 37 classes. We compare FuseNet-SF5, FuseNet-DF1 to the network trained with stacked RGB-D input.

	wall	floor	cabin	bed	chair	sofa	table	door	wdw	bslf	pic	cnter	blinds
RGB-D	69.46	86.10	35.56	58.29	60.02	43.09	46.37	27.76	43.30	19.70	36.24	25.48	29.11
SF5	**74.94**	**87.41**	**41.70**	**66.53**	**64.45**	**50.36**	**49.01**	**33.35**	**44.77**	**28.12**	**46.84**	**27.73**	**31.47**
DF1	69.48	86.09	35.57	58.27	60.03	43.09	46.38	27.78	43.31	19.75	36.30	25.44	29.12
	desk	shelf	ctn	drssr	pillow	mirror	mat	clths	ceil	books	fridge	tv	paper
RGB-D	10.19	5.34	43.02	23.93	30.70	31.00	0.00	**17.67**	63.10	21.79	22.69	31.31	12.05
SF5	**18.31**	**9.20**	**52.68**	**34.61**	**37.77**	**38.87**	0.00	16.67	**67.34**	**27.29**	**31.31**	31.64	16.01
DF1	15.61	7.44	42.24	28.74	31.99	34.73	0.00	15.82	60.09	24.28	23.63	**37.67**	**16.45**
	towel	shwr	box	board	person	stand	toilet	sink	lamp	btub	bag	mean	
RGB-D	13.21	4.13	14.21	40.43	10.00	11.79	59.17	45.85	26.06	51.75	12.38	31.95	
SF5	**16.55**	**6.06**	**15.77**	49.23	14.59	**19.55**	**67.06**	**54.99**	**35.07**	**63.06**	9.52	**37.29**	
DF1	13.60	1.54	15.47	**45.21**	**15.49**	17.46	63.38	48.09	27.06	56.85	**12.92**	34.02	

the three network architectures give very comparable results. However, for IoU scores, SF5 outperforms in 30 out of 37 classes in comparison to other two networks. Since the classwise IoU is a better measurement over global and mean accuracy, FuseNet obtains significant improvements over the network trained with stacked RGB-D, showing that depth fusion is a better approach to extract informative features from depth and to combine them with color features. In Figure 5, we demonstrate some visual comparison of the FuseNet.

5 Conclusions

In this paper, we have presented a fusion-based CNN network for semantic labeling on RGB-D data. More precisely, we have proposed a solution to fuse depth information with RGB data by making use of a CNN. The proposed network has an encoder-decoder type architecture, where the encoder part is composed of two branches of networks that simultaneously extract features from RGB and depth channels. These features are then fused into the RGB feature maps as the network goes deeper.

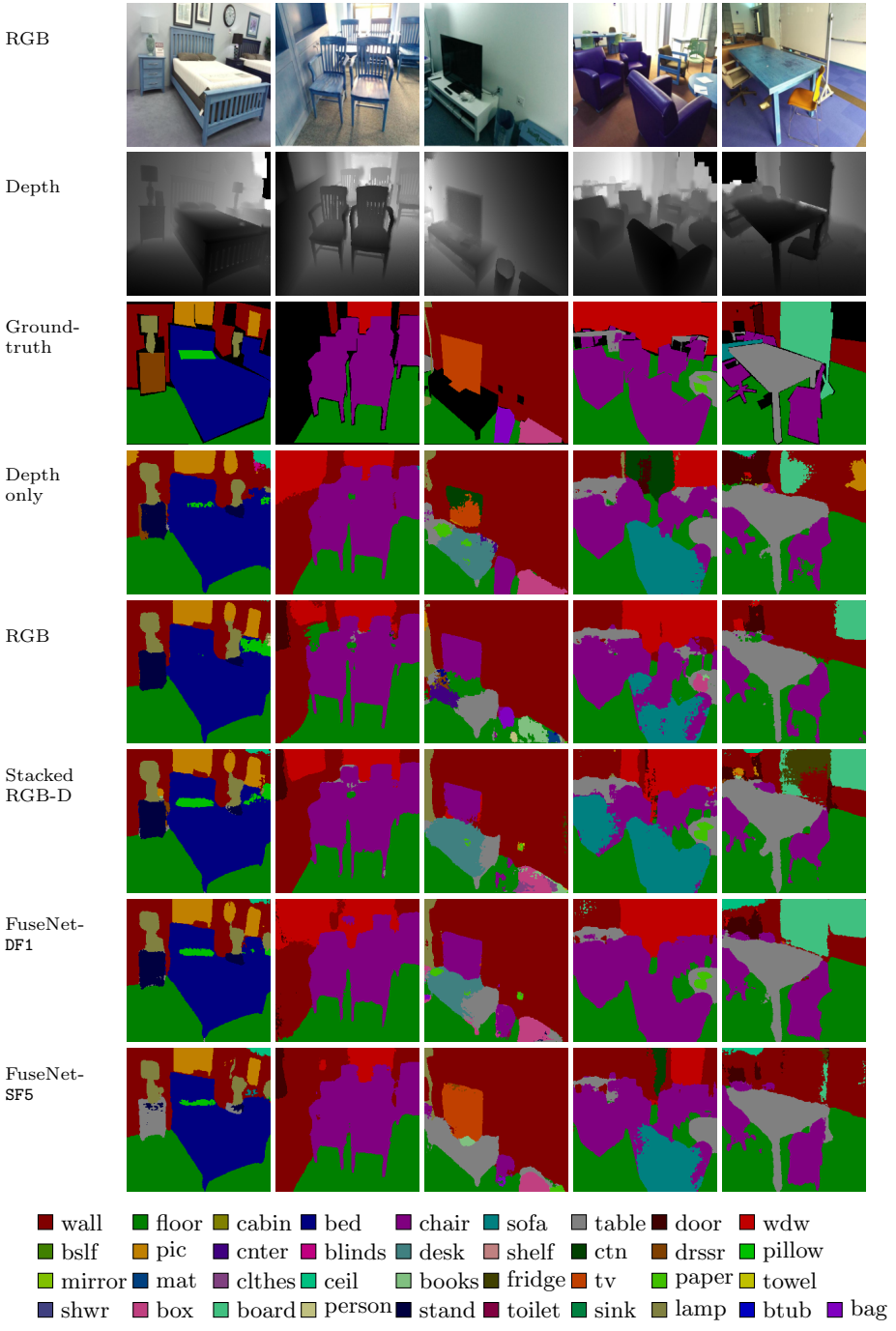


Fig. 5: Qualitative segmentation results for different architectures. The first three rows contain RGB and depth images along with the ground-truth, respectively, followed by the segmentation results. Last two rows contain the results obtained by our `DF1` and `SF5` approaches.

By conducting a comprehensive evaluation, we may conclude that the our approach is a competitive solution for semantic segmentation on RGB-D data. The proposed FuseNet outperforms the current CNN-based networks on the challenging SUN RGB-D benchmark [10]. We have also investigated two possible fusion approaches, *i.e.* dense fusion and sparse fusion. By applying the latter one with a single fusion operation we have obtained a slightly better performance. Nevertheless we may conclude that both fusion approaches provide similar results. Interestingly, we can also claim that HHA representation itself provides a superficial improvement to the depth information.

We also remark that a straight-forward extension of the proposed approach can be applied for other classification tasks such as image or scene classification.

Acknowledgement. This work was partially supported by the Alexander von Humboldt Foundation.

References

1. Gupta, S., Girshick, R., Arbelaez, P., Malik, J.: Learning rich features from RGB-D images for object detection and segmentation. In Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., eds.: Proceedings of European Conference on Computer Vision. Volume 8695 of Lecture Notes in Computer Science., Zurich, Switzerland, Springer (2014) 345–360
2. Pinheiro, P.O., Collobert, R.: Recurrent convolutional neural networks for scene labeling. In: Proceedings of International Conference on Machine Learning, Beijing, China (2014)
3. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, IEEE (2015) 3431–3440
4. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: Proceedings of IEEE International Conference on Computer Vision, Santiago, Chile, IEEE (2015) 1529–1537
5. Byeon, W., Breuel, T.M., Raue, F., Liwicki, M.: Scene labeling with LSTM recurrent neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, IEEE (2015) 3547–3555
6. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. Proceedings of IEEE International Conference on Computer Vision (2015)
7. Lin, G., Shen, C., van den Hengel, A., Reid, I.: Exploring context with deep structured models for semantic segmentation. arXiv preprint arXiv:1603.03183 (2016)
8. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: Proceedings of International Conference on Learning Representations, San Diego, CA, USA (2015)
9. Couprie, C., Farabet, C., Najman, L., LeCun, Y.: Indoor semantic segmentation using depth information. In: Proceedings of International Conference on Learning Representations. (2013)

10. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2015) 567–576
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Proceedings of International Conference on Learning Representations (2015)
12. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., eds.: Proceedings of European Conference on Computer Vision. Volume 8689 of Lecture Notes in Computer Science., Zurich, Switzerland, Springer (2014) 818–833
13. Badrinarayanan, V., Handa, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. arXiv preprint arXiv:1505.07293 (2015)
14. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680 (2015)
15. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15** (2014) 1929–1958
16. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. Computing Research Repository (2015)
17. Li, L.Z., Yukang, G., Xiaodan, L., Yizhou, Y., Hui, C., Liang, L.: RGB-D Scene labeling with long short-term memorized fusion model. arXiv preprint arXiv:1604.05000v2 (2016)
18. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F.R., Blei, D.M., eds.: Proceedings of International Conference on Machine Learning. Volume 37 of JMLR Proceedings., JMLR.org (2015) 448–456
19. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of IEEE International Conference on Computer Vision. (2015) 2650–2658
20. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
21. Bottou, L.: Stochastic gradient descent tricks. In: Neural Networks: Tricks of the Trade. Springer (2012) 421–436
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision **115** (2015) 211–252