

# Multi-View Deep Learning for Consistent Semantic Mapping with RGB-D Cameras

Lingni Ma<sup>1</sup>, Jörg Stückler<sup>2</sup>, Christian Kerl<sup>1</sup> and Daniel Cremers<sup>1</sup>

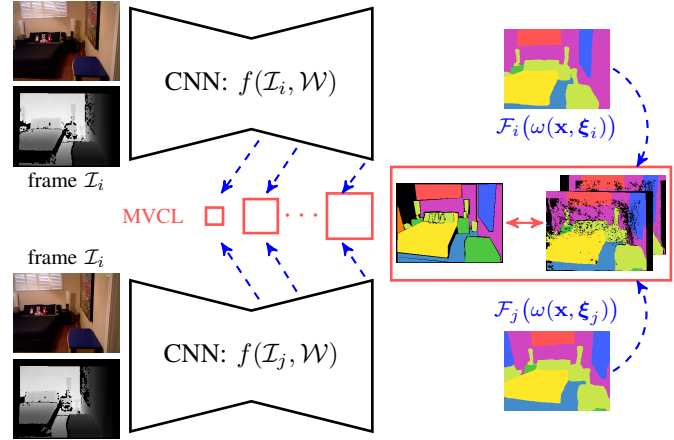
**Abstract**—Visual scene understanding is an important capability that enables robots to purposefully act in their environment. In this paper, we propose a novel approach to object-class segmentation from multiple RGB-D views using deep learning. We train a deep neural network to predict object-class semantics that is consistent from several view points in a semi-supervised way. At test time, the semantics predictions of our network can be fused more consistently in semantic keyframe maps than predictions of a network trained on individual views. We base our network architecture on a recent single-view deep learning approach to RGB and depth fusion for semantic object-class segmentation and enhance it with multi-scale loss minimization. We obtain the camera trajectory using RGB-D SLAM and warp the predictions of RGB-D images into ground-truth annotated frames in order to enforce multi-view consistency during training. At test time, predictions from multiple views are fused into keyframes. We propose and analyze several methods for enforcing multi-view consistency during training and testing. We evaluate the benefit of multi-view consistency training and demonstrate that pooling of deep features and fusion over multiple views outperforms single-view baselines on the NYUDv2 benchmark for semantic segmentation. Our end-to-end trained network achieves state-of-the-art performance on the NYUDv2 dataset in single-view segmentation as well as multi-view semantic fusion.

## I. INTRODUCTION

Intelligent robots require the ability to understand their environment through parsing and segmenting the 3D scene into meaningful objects. The rich appearance-based information contained in images renders vision a primary sensory modality for this task.

In recent years, large progress has been achieved in semantic object-class segmentation of images. Most current state-of-the-art approaches apply deep learning for this task. With RGB-D cameras, appearance as well as shape modalities can be combined to improve the semantic segmentation performance. Less explored, however, is the usage and fusion of multiple views onto the same scene which appears naturally in the domains of 3D reconstruction and robotics. Here, the camera is moving through the environment and captures the scene from multiple view points. Semantic SLAM aims at aggregating several views in a consistent 3D geometric and semantic reconstruction of the environment.

In this paper, we propose a novel approach for using multi-view context for deep learning of semantic segmentation of RGB-D images. We base our network architecture on a



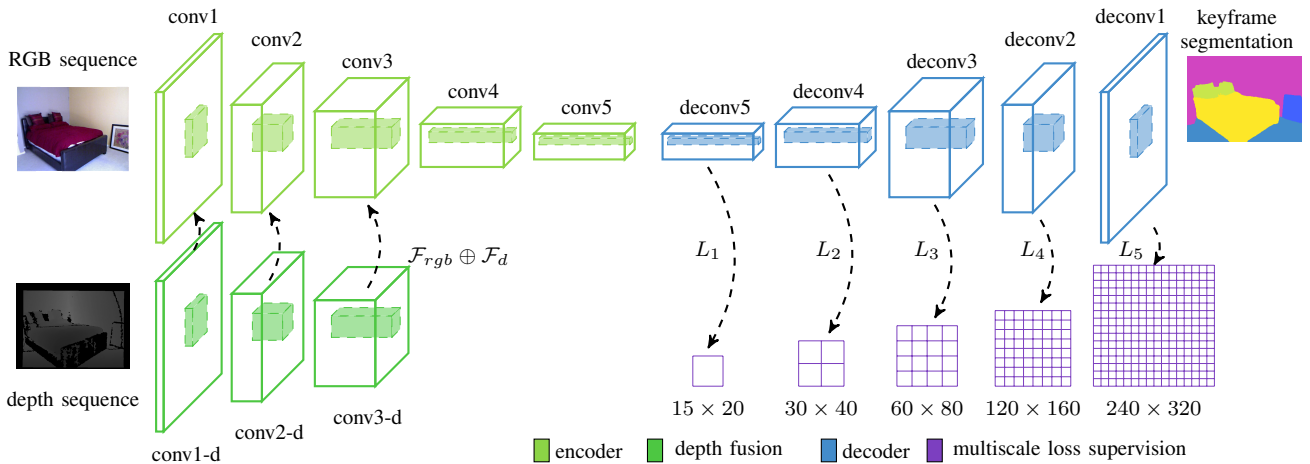
**Fig. 1:** We train CNN to predict multi-view consistent semantic segmentations for RGB-D images. The key innovations are the multi-view consistency layers (MVCL), which warp semantic prediction or feature maps at multiple scales into a common reference view based on the SLAM trajectory. Our approach improves performance for single-view segmentation and is specifically beneficial for multi-view fused segmentation at test time.

recently proposed deep convolutional neural network (CNN) for RGB and depth fusion [1] and enhance the approach with multi-scale loss minimization. Based on the trajectory estimate obtained through RGB-D simultaneous localization and mapping (SLAM), we train our CNN to predict multi-view consistent semantics in individual images. We propose and evaluate several approaches for enforcing multi-view consistency during training. A shared principle in our approaches is to use the SLAM trajectory estimate for warping network outputs or feature maps from nearby frames to keyframes with ground truth annotations. By this, the network not only learns features that are invariant under view-point change. Our semi-supervised training approach also makes better use of the annotated ground truth data than single-view learning. This alleviates the need for large amounts of annotated training data which is expensive to obtain for real imagery. Complementary to our training approach, we aggregate the predictions of our trained network in consistent semantic segmentations of keyframes at test time. The predictions of nearby overlapping images along the camera trajectory are fused into the keyframe based on the SLAM estimate in a probabilistic way.

In experiments, we evaluate the performance gain achieved through multi-view training and fusion at test time over single-view approaches. Our results demonstrate that multi-view max-pooling of feature maps during training best supports multi-view fusion at test time. Overall we find that

<sup>1</sup> Lingni Ma, Christian Kerl and Daniel Cremers are with the Computer Vision Group, Department of Computer Science, Technical University of Munich {lingni, kerl, cremers}@in.tum.de

<sup>2</sup> Jörg Stückler is with the Computer Vision Group, Visual Computing Institute, RWTH Aachen University, stueckler@vision.rwth-aachen.de



**Fig. 2:** The CNN encoder-decoder architecture used in our approach. Input to our network are RGB-D images. The network extracts features from depth images in a separate encoder whose features are fused with RGB features in a fused encoder network. The encoded features at the lowest resolution are successively refined through deconvolutions in a decoder. To guide the refinement, we train the network in a deeply-supervised manner in which segmentation loss is computed at all scales of the decoder.

enforcing multi-view consistency during training improves fusion at test time significantly over fusing predictions from networks trained on single views. Our end-to-end trained network achieves state-of-the-art performance on the NYUDv2 dataset in single-view segmentation as well as multi-view semantic fusion. While the fused keyframe segmentation can be directly used in robotic perception, our approach can also be useful as a building block for geometric and semantic SLAM using RGB-D cameras.

## II. RELATED WORK

Recently, remarkable progress has been achieved in semantic image segmentation using deep neural networks and, in particular, CNNs. On many benchmarks, these approaches excel previous techniques by a great margin. As one early attempt, Couprie et al. [2] propose a multiscale CNN architecture to combine information at different perceptive field resolutions and achieved reasonable segmentation results. They were also one of the first to train a CNN with depth information for RGB-D image segmentation. Gupta et al. [3] integrate depth into the R-CNN approach by Girshick et al. [4] to detect objects in RGB-D images. To apply a CNN pretrained on ImageNet on the depth images, they propose to transform depth into a disparity, height and angle encoding. For semantic segmentation, they train a classifier to label superpixels based on the CNN features. Long et al. [5] propose a fully convolutional network (FCN) which enables end-to-end training of a deep CNN for semantic segmentation. Their FCN architecture reduces the input’s spatial resolution by a great factor through layers of filtering and pooling. It fuses low with high resolution predictions to obtain the final prediction. Inspired by FCN and auto-encoder networks [6], encoder-decoder architectures have been proposed for semantic segmentation [7]. For RGB-D images, Eigen et al. [8] propose multi-task CNN training that aims to predict depth, surface normals and semantics with one uniform network and achieve very good performance.

FuseNet [1] proposes a principled approach for fusing RGB and depth cues in a single encoder-decoder CNN trained end-to-end for semantic image segmentation. Li et al. [9] use a LSTM recurrent neural network to fuse RGB and depth cues and obtain smooth predictions. Lin et al. [10] design a CNN that corresponds to a conditional random field (CRF) and use piecewise training to learn both unary and pairwise potentials end-to-end. While this method produces very good results, it requires a mean-field approximation for coarse inference and high-resolution refinement using a dense CRF [11]. Our approach trains a network on multi-view consistency and fuses the results from multiple view points. It is complementary to the above mentioned single-view CNN approaches.

In the domain of semantic SLAM, Salas-Moreno et al. [12] developed the SLAM++ algorithm to perform RGB-D tracking and mapping at the object instance level. This method works well for indoor scenes which contain many repeated objects with predefined CAD models in a database. Hermans et al. [13] proposed 3D semantic mapping for indoor RGB-D sequences based on RGB-D visual odometry and a random forest classifier that performs semantic image segmentation. The individual frame segmentations are projected into 3D and a dense CRF [11] on the point cloud smoothes the semantic segmentation in 3D. Stückler et al. [14] perform RGB-D SLAM and probabilistically fuse the semantic segmentations of individual frames obtained with a random forest in multi-resolution voxel maps. Recently, Armeni et al. [15] propose a hierarchical parsing method for large-scale 3D point clouds of indoor environments. They first separate point clouds into disjoint spaces, *i.e.*, single rooms, and then further cluster points at the object level according to handcrafted features.

In contrast to the popularity of CNNs for image-based segmentation, it is less common to apply CNNs for semantic segmentation on multi-view 3D reconstructions. This

is partially due to the lack of an organized structure in point clouds or the less manageable scale of volumetric representations for training a deep neural network. Recently, Riegler et al. [16] apply 3D CNNs on sparse octree data structures to perform semantic segmentation on voxels. Nevertheless, the volumetric representations may discard details through the voxelization which are present at the original image resolution. McCormac et al. [17] proposed to fuse CNN semantic image segmentations on a 3D surfel map [18]. In contrast to our approach, this method does not use multi-view consistency during CNN training and cannot leverage the view-point invariant features learned by our network. Closely related to our approach for enforcing multi-view consistency is the approach by Su et al. [19] who investigate the task of 3D shape recognition. They render multiple views onto 3D shape models which are fed into a CNN feature extraction stage that is shared across views. The features are max-pooled across view-points and fed into a second CNN stage that is trained for shape recognition. Our approach uses multi-view pooling for the task of semantic segmentation and is trained using realistic imagery and SLAM pose estimates. Our trained network is able to classify single views, but we demonstrate that multi-view fusion using the network trained on multi-view consistency improves segmentation performance over single-view trained networks.

### III. CNN APPROACH TO SEMANTIC RGB-D IMAGE SEGMENTATION

In this section, we detail our CNN architecture for semantic segmentation in RGB-D images. We base our approach on FuseNet [1] which consistently fuses RGB and depth images for semantic segmentation, and enhance the approach with multi-scale loss minimization. By this, we already achieve a significant improvement in semantic segmentation performance on single views.

#### A. Network Architecture

Fig. 2 illustrates our CNN architecture for semantic image segmentation in RGB-D images. The network follows an encoder-decoder design, similar to previous work on semantic segmentation [7]. The encoder extracts a hierarchy of features through convolutional layers and aggregates spatial information by pooling over a local neighborhood to increase the perceptive field. The last layer of the encoder outputs high dimensional feature maps with low spatial resolution. The decoder then upsamples the low-resolution feature maps through several layers of memorized unpooling and deconvolution and successively refines the low-resolution feature maps back to the input resolution. To learn features from RGB-D images, we adopt the FuseNet architecture [1] which is shown to be more efficient in learning features from RGB-D images in comparison to simple concatenation of RGB and depth or to the use of HHA [3] representation. As demonstrated in Fig. 2, the network contains two branches each learning features from RGB ( $\mathcal{F}_{rgb}$ ) and depth ( $\mathcal{F}_d$ ), respectively. The feature maps from the depth branch are

consistently fused into the RGB branch at each scale. We denote the fusion of the feature maps by  $\mathcal{F}_{rgb} \oplus \mathcal{F}_d$ .

For semantic segmentation, the label set is denoted as  $\mathcal{L} = \{1, 2, \dots, K\}$  and the category index is indicated with subscript  $j$ . Following notation convention, we compute the classification score  $\mathcal{S} = (s_1, s_2, \dots, s_K)$  at spatial location  $\mathbf{x}$  and map it to the probability distribution  $\mathcal{P} = (p_1, p_2, \dots, p_K)$  with the softmax function  $\sigma(\cdot)$ . Network inference obtains the probability

$$p_j(\mathbf{x}, \mathcal{W} | \mathcal{I}) = \sigma(s_j(\mathbf{x}, \mathcal{W})) = \frac{\exp(s_j(\mathbf{x}, \mathcal{W}))}{\sum_k^K \exp(s_k(\mathbf{x}, \mathcal{W}))}, \quad (1)$$

of all pixels  $\mathbf{x}$  in the image for being labelled as class  $j$ , given input RGB-D image  $\mathcal{I}$  and network parameters  $\mathcal{W}$ .

#### B. Multi-Scale Loss Minimization

We use the cross-entropy loss to learn network parameters for semantic segmentation from ground-truth annotations  $l_{gt}$ ,

$$L(\mathcal{W}) = -\frac{1}{N} \sum_i^N \sum_j^K \mathbb{I}[j = l_{gt}] \log p_j(\mathbf{x}_i, \mathcal{W} | \mathcal{I}), \quad (2)$$

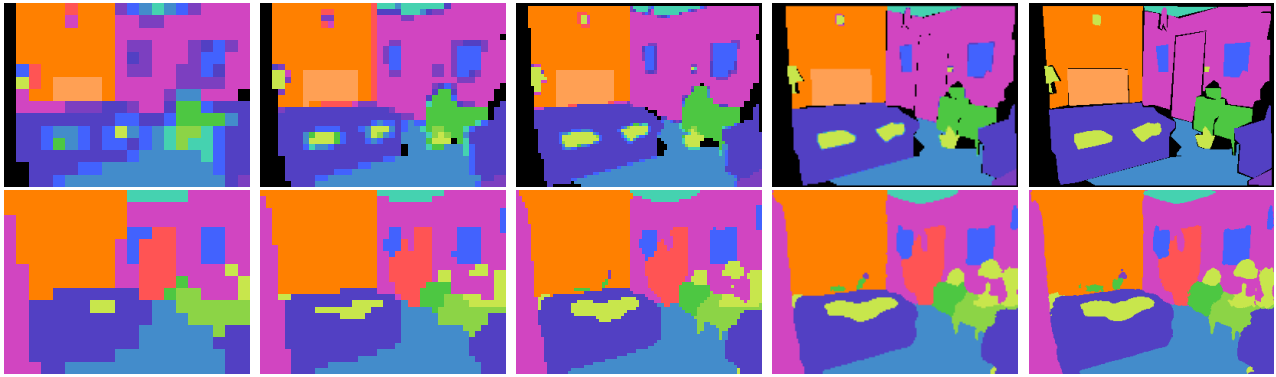
where  $N$  is the number of pixels. This loss minimizes the Kullback-Leibler (KL) divergence between predicted distribution and the ground-truth, assuming the ground-truth has a one-hot distribution on the true label.

The encoder of our network contains five pooling layers of  $2 \times 2$  filter size and downsamples the input resolution by a factor of 32. The decoder learns to refine the low resolution back to the original with five scales of memorized unpooling layers followed by deconvolution. In order to guide the decoder through the successive refinement, we adopt a deeply supervised learning method [20], [21] and compute the cross-entropy loss at all upsample scales. To this end, we append a classification layer at each deconvolution scale and compute the loss for the respective resolution ground-truth which is obtained through stochastic pooling [22] of the full resolution annotation (see Fig. 3 for an example).

### IV. MULTI-VIEW CONSISTENCY

The key innovation of this work is to explore the use of temporal multi-view consistency within an RGB-D sequence for CNN training and prediction. While convolutional neural networks (CNN) have been shown to obtain the state-of-the-art semantic segmentation performances for many datasets, most of these studies focus on single views. When observing a scene from a moving camera such as on a mobile robot, the system obtains multiple different views onto the same objects. We aim to use this for increasing the consistency of semantic maps by fusing semantic image segmentations in keyframes from multiple view points. Moreover, we can make use of the multi-view information in RGB-D video for training a CNN to produce consistent semantic segmentations under view-point changes.

We define each training sequence to contain one reference keyframe  $\mathcal{I}_k$  with ground-truth semantic annotation and several nearby overlapping frames  $\mathcal{I}_i$ . The relative poses  $\xi$  of the nearby frames towards the reference keyframe are



**Fig. 3:** Example of multi-scale ground-truth and predictions. Upper row: successive subsampled of ground-truth annotation obtained through stochastic pooling. Lower row: CNN prediction on each scale. The resolutions are coarse to fine from left to right with  $20 \times 15$ ,  $40 \times 30$ ,  $80 \times 60$ ,  $160 \times 120$  and  $320 \times 240$ .

estimated through a SLAM method such as [23]. In order to impose temporal consistency, we adopt the warping concept from multiview geometry to associate pixels between view points. To this end, we introduce warping layers into the CNN architecture that synthesize the CNN output at any stage in one view point by sampling the output of another view point based on the SLAM pose estimate. These layers can be seen as a variant of spatial transformers [24]. Through these warping layers, it is possible to impose temporal multi-view consistency. In the following, we describe our warping layers and introduce several variants of multi-view consistency constraints based on warping.

#### A. Multiview Association Through Warping

Given the normalized 2D image coordinate  $\mathbf{x} \in \mathbb{R}^2$ , its warped image location

$$\mathbf{x}^\omega := \omega(\mathbf{x}, \boldsymbol{\xi}) = \pi(\mathbf{T}(\boldsymbol{\xi}) \pi^{-1}(\mathbf{x}, Z_i(\mathbf{x}))) \quad (3)$$

is determined through the warping function  $\omega(\mathbf{x}, \boldsymbol{\xi})$  which transforms the location from one camera frame to the other based on the depth  $Z_i(\mathbf{x})$  at  $\mathbf{x}$  in image  $\mathcal{I}_i$  and the SLAM pose estimate  $\boldsymbol{\xi}$ . The functions  $\pi$  and its inverse  $\pi^{-1}$  project homogeneous 3D coordinates to normalized image coordinates and vice versa, while  $\mathbf{T}(\boldsymbol{\xi})$  denotes the homogeneous transformation matrix for pose  $\boldsymbol{\xi}$ .

The warping function associates pixels between two view-points. Using this association, it is possible to synthesize the output of any CNN layer in one view point by sampling the output of another view point. For a network with several spatial resolutions, the warping grid only needs to be computed once at the input resolution. For this purpose, we normalize the warping grid by the input resolution to obtain a canonical representation within the range of  $[-1, 1]$ . The canonical representation enables efficient generation of warping grids at any lower resolution through average pooling layers. Using bilinear interpolation, it is then straight-forward to synthesize the output at any scale and gradients can be back-propagated through the warping layer. With a slight abuse of notation, we denote the operation of synthesizing the layer output  $\mathcal{F}$  given the warping by  $\mathcal{F}^\omega := \mathcal{F}(\omega(\mathbf{x}, \boldsymbol{\xi}))$ .

#### B. Consistency Through Warp Augmentation

One way to enforce multi-view consistency in the segmentation is to warp the predictions of nearby frames into the ground-truth annotated keyframe and computing a supervised loss there. This approach can be interpreted as a kind of data augmentation using the available nearby frames.

We implement this consistency method by warping the keyframe into the nearby frame, and synthesize the classification score of the nearby frame from the keyframe’s view point. We compute the cross-entropy loss on this synthesized semantic segmentation. Within RGB-D sequences, objects can appear at various scales, image locations and view angles. Propagating the keyframe annotation into the other frames implicitly regulates the network predictions to be invariant under these transformations.

#### C. Consistency Through Bayesian Fusion

Given a sequence of measurements and predictions at test time, Bayesian fusion is frequently applied to aggregate the semantic segmentations of individual views. Without a loss of generality, let us denote the semantic labelling of a pixel by  $y$  and its measurement in frame  $i$  by  $z_i$ . We use the notation  $z^i$  for the set of measurements up to frame  $i$ . According to Bayes rule,

$$p(y | z^i) = \frac{p(z_i | y, z^{i-1}) p(y | z^{i-1})}{p(z_i | z^{i-1})} \quad (4)$$

$$= \eta_i p(z_i | y, z^{i-1}) p(y | z^{i-1}) \quad (5)$$

Suppose measurements satisfy the *i.i.d.* condition, i.e.  $p(z_i | y, z^{i-1}) = p(z_i | y)$ , and equal a-priori probability for each class, then Equation (4) simplifies to

$$p(y | z^i) = \eta_i p(z_i | y) p(y | z^{i-1}) = \prod_i \eta_i p(z_i | y). \quad (6)$$

Put simple, Bayesian fusion is implemented by taking the product over the individual frame semantic labelling likelihoods at a pixel and normalizing the product to yield a valid probability distribution. This process can also be implemented recursively on a sequence of frames.

When training our CNN for multi-view consistency using Bayesian fusion, we warp the predictions of nearby frames

into the keyframe using the SLAM pose estimate. We obtain the fused prediction at each keyframe pixel by summing the unnormalized log labelling likelihoods instead of the individual frame softmax outputs. Applying softmax on the sum of log labelling likelihoods yields the fused labelling probability distribution. This method is equivalent to Equation (6) since

$$\frac{\prod_i p_{i,j}^\omega}{\sum_k \prod_i p_{i,k}^\omega} = \frac{\prod_i \sigma(s_{i,j}^\omega)}{\sum_k \prod_i \sigma(s_{i,k}^\omega)} = \sigma\left(\sum_i s_{i,j}^\omega\right), \quad (7)$$

where  $s_{i,j}^\omega$  and  $p_{i,j}^\omega$  denote the warped classification scores and probabilities, respectively, and  $\sigma(\cdot)$  is the softmax function as defined in Equation (1).

#### D. Consistency Through Multi-View Max-Pooling

While Bayesian fusion provides an approach to integrate several measurements in the probability space, we also explore direct fusion in the feature space using multi-view max-pooling of the warped feature maps. We warp the feature maps preceding the classification layers at each scale in our decoder into the keyframe and apply max-pooling over corresponding feature activations at the same warped location to obtain a pooled feature map in the keyframe,

$$\mathcal{F} = \text{max\_pool}(\mathcal{F}_1^\omega, \mathcal{F}_2^\omega, \dots, \mathcal{F}_N^\omega). \quad (8)$$

The fused feature maps are classified and the resulting semantic segmentation is compared to the keyframe ground-truth for loss calculation.

## V. EVALUATION

We evaluate our proposed approach using the NYUDv2 [25] RGB-D dataset. The NYUDv2 dataset provides 1449 pixel-wise annotated RGB-D images that is commonly split into a subset of 795 frames for training/validation (trainval) and 654 frames for testing. The dataset contains various indoor environments captured with consumer RGB-D cameras. The original sequences that contain these 1449 images are also available. Using the DVO-SLAM algorithm [23], we determine the camera motion around each annotated keyframe to obtain training and test sequences. As a result, we obtain sequences with in total 267,675 frames, despite that tracking fails for 30 out of 1449 keyframes. Following the original trainval/test split, we use 770 sequences consisting of 143,670 total frames for training and 649 sequences with 124,005 frames for testing. For benchmarking, we evaluate our method for the NYUDv2 13-class [2] and 40-class [26] semantic segmentation tasks. For training and testing, we use the raw depth images without inpainted missing values. While larger RGB-D datasets with ground-truth annotation are available, unfortunately they do not provide sequences.

#### A. Training Details

We implemented our approach using the Caffe framework [27]. For all experiments, the network parameters are initialized as follows. For the convolutional kernels in the encoder, we use the pretrained 16-layer VGGNet model [28] and for the deconvolutional kernels in the decoder, we use

**TABLE I:** Single-view semantic segmentation accuracy of our network in comparison to the state-of-the-art methods for NYUDv2 13-class and 40-class segmentation tasks.

	methods	input	pixelwise	classwise	IoU
NYUDv2 13 classes	Coupric et al. [2]	RGB-D	52.4	36.2	-
	Hermans et al. [13]	RGB-D	54.2	48.0	-
	SceneNet [31]	DHA	67.2	52.5	-
	Eigen et al. [8]	RGB-D-N	75.4	66.9	52.6
	FuseNet-SF3 [1]	RGB-D	75.8	66.2	54.2
	MVCNet-Mono	RGB-D	77.6	68.7	56.9
	MVCNet-Augment	RGB-D	77.6	69.3	57.2
	MVCNet-Bayesian	RGB-D	<b>77.8</b>	<b>69.4</b>	<b>57.3</b>
	MVCNet-MaxPool	RGB-D	77.7	<b>69.5</b>	<b>57.3</b>
	NYUDv2 40 classes	RCNN [3]	RGB-HHA	60.3	35.1
FCN-16s [5]		RGB-HHA	65.4	46.1	34.0
Eigen et al. [8]		RGB-D-N	65.6	45.1	34.1
FuseNet-SF3 [1]		RGB-D	66.4	44.2	34.0
Context-CRF [10]		RGB	67.6	49.6	37.1
MVCNet-Mono		RGB-D	68.6	48.7	37.6
MVCNet-Augment		RGB-D	<b>68.6</b>	<b>49.9</b>	<b>38.0</b>
MVCNet-Bayesian		RGB-D	68.4	49.5	37.4
MVCNet-MaxPool		RGB-D	<b>69.1</b>	<b>50.1</b>	<b>38.0</b>

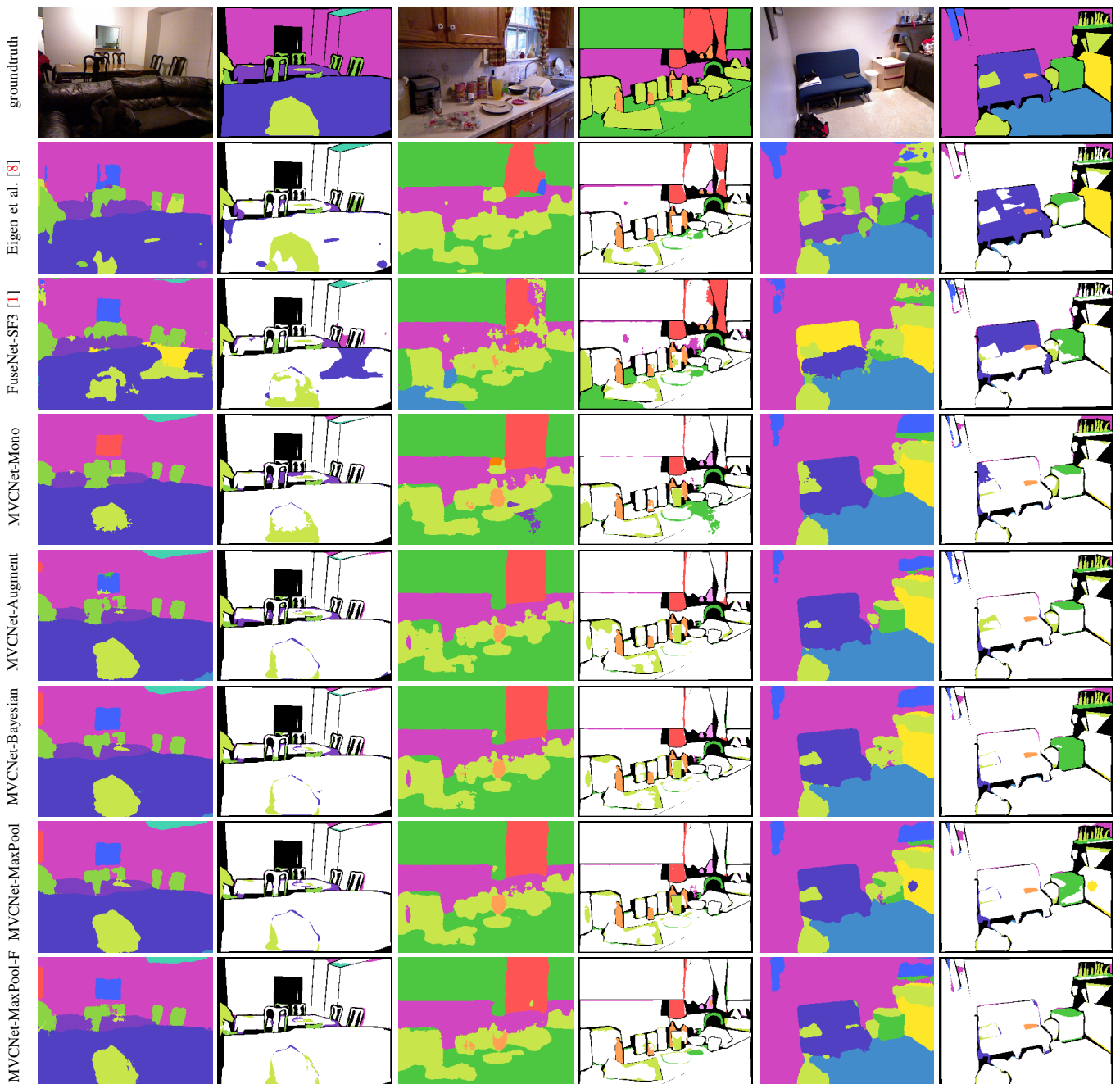
He initialization [29]. We train the network with stochastic gradient descent (SGD) [30] with 0.9 momentum, 0.0005 weight decay and set the batch size to 4. The learning rate is set to 0.001 and multiplied by a factor of 0.9 every 30,000 iterations. We apply random shuffling after each epoch and train the network until convergence. All the images are resized to a resolution of  $320 \times 240$  pixels as input to the network and the predictions are also up to this scale. We use cubic interpolation to downsample RGB images and nearest-neighbor interpolation to downsample depth and label images. Most of the keyframes have long tracking sequences, where tracking drift typically accumulates along the sequence. Hence for multi-view training, we feed the close-by frames first to the network and gradually include 10 further-away frames in 5 epochs.

#### B. Evaluation Criteria

We measure the semantic segmentation performance of our network with three criteria: global pixelwise accuracy, average classwise accuracy and average intersection-over-union (IoU) scores. These three criteria can be calculated from the confusion matrix  $\mathbf{C} \in \mathbb{R}^{K \times K}$ . Each element in the confusion matrix  $c_{ij}$  is the total amount of pixels belonging to class  $i$  which are predicted to be class  $j$ . The global pixelwise accuracy is computed by  $\sum_i c_{ii} / \sum_{ij} c_{ij}$  and the average classwise accuracy is computed by  $\frac{1}{K} \sum_i (c_{ii} / \sum_j c_{ij})$ . The average IoU score is calculated according to  $\frac{1}{K} \sum_i (c_{ii} / (\sum_i c_{ij} + \sum_j c_{ij} - c_{ii}))$ .

#### C. Single Frame Segmentation

In a first set of experiments, we evaluate the performance of several variants of our network for direct semantic segmentation of frames. This means we do not fuse predictions from nearby frames to obtain the final prediction in frame. We predict semantic segmentation with our trained models on the 654 test images of the NYUDv2 dataset and compare



**Fig. 4:** Qualitative semantic segmentation results of our methods and several state-of-the-art baselines on NYUDv2 13-class segmentation (see Table III for color coding, left columns: semantic segmentation, right columns: falsely classified pixels, black is void). Our multi-view consistency trained models produce more accurate and homogeneous results than single-view methods. Bayesian fusion further improves segmentation quality (e.g. MVCNet-MaxPool-F).

our methods with state-of-art approaches. The results are shown in Table I. Unless otherwise stated, we take the results from the original papers for comparison and report their best results (i.e. SceneNet-FT-NYU-DO-DHA model for SceneNet [31], VGG-based model for Eigen et al. [8]). The result of Hermans et al. [13] is obtained after applying a dense CRF [11] for each image and in-between neighboring 3D points to further smoothen their results. We also remark that the results reported here for the Context-CRF model

are finetuned on NYUDv2 like in our approach to facilitate comparison. Furthermore, the network output is refined using a dense CRF [11] which is claimed to increase the accuracy of the network by approximately 2%. The results for FuseNet-SF3 are obtained by our own implementation. Our baseline model MVCNet-Mono is trained without multi-view consistency, which amounts to FuseNet with multiscale deeply supervised loss at decoder. However, we apply single image augmentation to train the FuseNet-SF3 and MVCNet-

**TABLE II:** Multi-view segmentation accuracy of our network using Bayesian fusion for NYUDv2 13-class and 40-class segmentation.

	methods	pixelwise	classwise	IoU
NYUDv2 13 classes	FuseNet-SF3 [1]	77.19	67.46	56.01
	MVCNet-Mono	78.70	69.61	58.29
	MVCNet-Augment	78.94	70.48	58.93
	MVCNet-Bayesian	<b>79.13</b>	70.48	59.04
	MVCNet-MaxPool	<b>79.13</b>	<b>70.59</b>	<b>59.07</b>
NYUDv2 40 classes	FuseNet-SF3 [1]	67.74	44.92	35.36
	MVCNet-Mono	70.03	49.73	39.12
	MVCNet-Augment	70.34	51.73	<b>40.19</b>
	MVCNet-Bayesian	70.24	51.18	39.74
	MVCNet-MaxPool	<b>70.66</b>	<b>51.78</b>	40.07

Mono with random scaling between  $[0.8, 1.2]$ , random crop and mirror. This data augmentation is not used for multi-view training. Nevertheless, our results show that the different variants of multi-view consistency training outperform the state-of-art methods for single image semantic segmentation. Overall, multi-view max-pooling (MVCNet-MaxPool) has a small advantage over the other multi-view consistency training approaches (MVCNet-Augment and MVCNet-Bayesian).

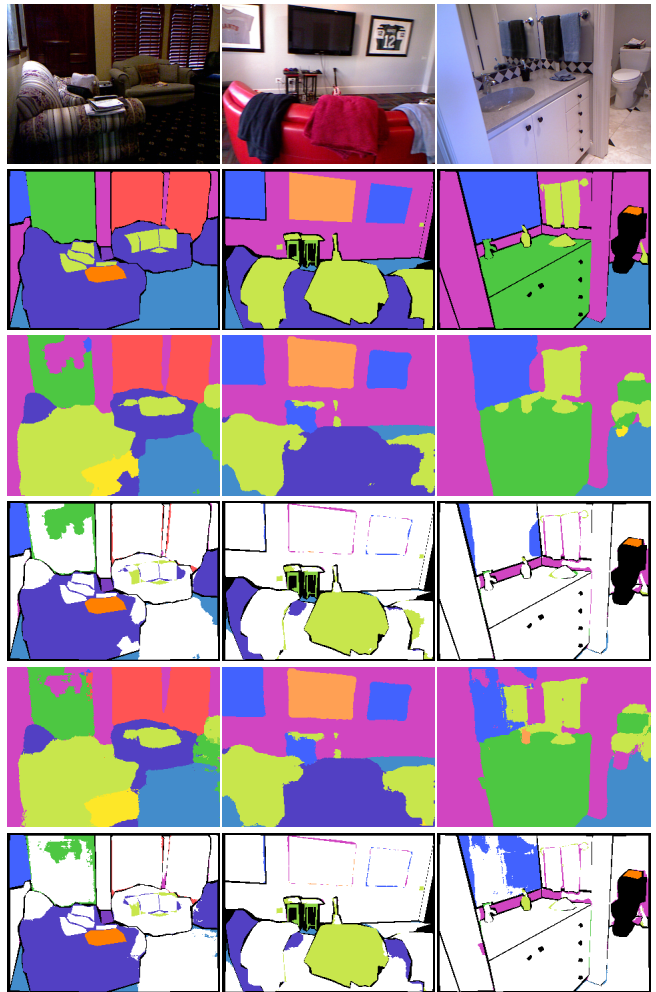
#### D. Multi-View Fused Segmentation

Since we train on sequences, in the second set of experiment, we also evaluate the fused semantic segmentation over the test sequences. The number of fused frames is fixed to 50, which are uniformly sampled over the entire sequence. Due to the lack of ground-truth for neighboring frames, we fuse the prediction of neighboring frames in the keyframes using Bayesian fusion according to Equation (7). This fusion is typically applied for semantic mapping using RGB-D SLAM. The results are shown in Table II. Bayesian multi-view fusion improves the semantic segmentation by approx. 2% on all evaluation measures towards single-view segmentation. Also, the training for multi-view consistency achieves a stronger gain over single-view training (MVCNet-Mono) when fusing segmentations compared to single-view segmentation. This performance gain is observed in the qualitative results in Fig. 4. It can be seen that our multi-view consistency training and Bayesian fusion produces more accurate classifications and more homogeneous segmentations. Fig. 5 shows typical challenging cases for our model.

We also compare classwise and average IoU scores for 13-class semantic segmentation on NYUDv2 in Table III. The results of Eigen et al. [8] are from their publicly available model tested on  $320 \times 240$  resolution. The results demonstrate that our approach gives high performance gains across all occurrence frequencies of the classes in the dataset.

## VI. CONCLUSION

In this paper we propose methods for enforcing multi-view consistency during the training of CNN models for semantic RGB-D image segmentation. We base our CNN design on FuseNet [1], a recently proposed CNN architecture in an encoder-decoder scheme for semantic segmentation of RGB-D images. We augment the network with multi-scale
















**Fig. 5:** Challenging cases for MVCNet-MaxPool-F (top to bottom: RGB image, ground-truth, single-view prediction on keyframe, multi-view prediction fused in keyframe). On the left, the network fails to classify the objects for all frames. In the middle, the network makes some errors in single-view prediction, but through multi-view fusion, some mistakes are corrected. On the right, multi-view fusion degenerates performance due to the mirror reflections.

loss supervision to improve its performance. We present and evaluate three different approaches for multi-view consistency training. Our methods use an RGB-D SLAM trajectory estimate to warp semantic segmentations or feature maps from one view point to another. Multi-view max-pooling of feature maps overall provides the best performance gains in single-view segmentation and fusion of multiple views.

We demonstrate the superior performance of multi-view consistency training and Bayesian fusion on the NYUDv2 13-class and 40-class semantic segmentation benchmark. All multi-view consistency training approaches outperform single-view trained baselines. They are key to boosting segmentation performance when fusing network predictions from multiple view points during testing. On NYUDv2, our model sets a new state-of-the-art performance using an end-to-end trained network for single-view predictions as well as multi-view fused semantic segmentation without further

**TABLE III:** NYUDv2 13-class semantic segmentation IoU scores. Our method achieves best per-class accuracy and average IoU.

method		 bed	 objects	 chair	 furniture	 ceiling	 floor	 decorat.	 sofa	 table	 wall	 window	 books	 TV	average accuracy
class frequency		4.08	7.31	3.45	12.71	1.47	9.88	3.40	2.84	3.42	24.57	4.91	2.78	0.99	
single-view	Eigen et al. [8]	56.71	38.29	50.23	54.76	64.50	89.76	45.20	47.85	42.47	74.34	56.24	45.72	34.34	53.88
	FuseNet-SF3 [1]	61.52	37.95	52.67	53.97	64.73	89.01	47.11	57.17	39.20	75.08	58.06	37.64	29.77	54.14
	MVCNet-Mono	65.27	37.82	54.09	59.39	65.26	89.15	49.47	57.00	44.14	75.31	57.22	49.21	36.14	56.88
	MVCNet-Augment	65.33	38.30	54.15	<b>59.54</b>	<b>67.65</b>	89.26	49.27	55.18	43.39	74.59	58.46	<b>49.35</b>	<b>38.84</b>	57.18
	MVCNet-Bayesian	<b>65.76</b>	38.79	<b>54.60</b>	59.28	67.58	89.69	48.98	<b>56.72</b>	42.42	75.26	<b>59.55</b>	49.27	36.51	57.26
MVCNet-MaxPool	65.71	<b>39.10</b>	54.59	59.23	66.41	<b>89.94</b>	<b>49.50</b>	56.30	<b>43.51</b>	<b>75.33</b>	59.11	49.18	37.37	<b>57.33</b>	
multi-view	FuseNet-SF3 [1]	64.95	39.62	55.28	55.90	64.99	89.88	47.99	<b>60.17</b>	42.40	76.24	59.97	39.80	30.91	56.01
	MVCNet-Mono	67.11	40.14	56.39	60.90	66.07	89.77	50.32	59.49	46.12	76.51	59.03	48.80	37.13	58.29
	MVCNet-Augment	68.22	40.04	56.55	61.82	67.88	90.06	50.85	58.00	<b>45.98</b>	75.85	60.43	50.50	<b>39.89</b>	58.93
	MVCNet-Bayesian	<b>68.38</b>	<b>40.87</b>	<b>57.10</b>	<b>61.84</b>	<b>67.98</b>	<b>90.64</b>	50.05	59.70	44.73	76.50	<b>61.75</b>	<b>51.01</b>	36.99	59.04
MVCNet-MaxPool	68.09	41.58	56.88	61.56	67.21	<b>90.64</b>	50.69	59.73	45.46	<b>76.68</b>	61.28	50.60	37.51	<b>59.07</b>	

postprocessing stages such as dense CRFs.

In future work, we want to further investigate integration of our approach in a semantic SLAM system, for example, through seamless coupling of pose tracking and SLAM with our semantic predictions.

## REFERENCES

- [1] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Proceedings of the 13th Asian Conference on Computer Vision (ACCV)*, 2016.
- [2] C. Couprie, C. Farabet, L. Najman, and Y. Lecun, *Indoor semantic segmentation using depth information*. April 2013.
- [3] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *European Conference on Computer Vision (ECCV)*, 2014.
- [4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [5] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Nov. 2015.
- [6] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19 (NIPS)*, pp. 153–160, 2007.
- [7] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pp. 1520–1528, 2015.
- [8] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [9] Z. Li, Y. Gan, X. Liang, Y. Yu, H. Cheng, and L. Lin, "LSTM-CF: Unifying context modeling and fusion with LSTMs for RGB-D scene labeling," in *European Conference on Computer Vision (ECCV)*, pp. 541–557, Springer, 2016.
- [10] G. Lin, C. Shen, A. van den Hengel, and I. D. Reid, "Exploring context with deep structured models for semantic segmentation," *CoRR*, vol. abs/1603.03183, 2016.
- [11] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," pp. 109–117, 2011.
- [12] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "SLAM++: Simultaneous localisation and mapping at the level of objects," *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [13] A. Hermans, G. Floros, and B. Leibe, "Dense 3D semantic mapping of indoor scenes from rgb-d images," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2631–2638, 2014.
- [14] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke, "Dense real-time mapping of object-class semantics from RGB-D video," *Journal of Real-Time Image Processing*, 2015.
- [15] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3D semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [16] G. Riegler, A. O. Ulusoy, and A. Geiger, "OctNet: Learning deep 3d representations at high resolutions," *CoRR*, vol. abs/1611.05009, 2016.
- [17] J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," *CoRR*, vol. abs/1609.05130, 2016.
- [18] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *Intl. J. of Robotics Research (IJRR)*, 2016.
- [19] H. Su, S. Maji, E. Kalogerakis, and E. G. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE Int. Conf. on Computer Vision (ICCV)*, 2015.
- [20] C. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. of the 18th Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2015.
- [21] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.
- [22] M. Zeiler and R. Fergus, *Stochastic pooling for regularization of deep convolutional neural networks*. 2013.
- [23] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for RGB-D cameras," in *Proc. of IROS*, 2013.
- [24] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28 (NIPS)* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 2017–2025, 2015.
- [25] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *European Conference on Computer Vision (ECCV)*, 2012.
- [26] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013.
- [27] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.
- [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [30] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, pp. 421–436, Springer, 2012.
- [31] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla, "Scenet: Understanding real world indoor scenes with synthetic data," in *IEEE Int. Conf. on Computer Vision (CVPR)*, 2016.