Learning Common Representation from RGB and Depth Images

Giorgio Giannone[1] and Boris Chidlovskii[2] *†

Abstract

We propose a new deep learning architecture for the tasks of semantic segmentation and depth prediction from RGB-D images. We revise the state of art based on the RGB and depth feature fusion, where both modalities are assumed to be available at train and test time. We propose a new architecture where the feature fusion is replaced with a common deep representation. Combined with an encoder-decoder network for feature map extraction, the architecture can jointly learn models for semantic segmentation and depth estimation based on their common representation. This representation, inspired by multi-view learning, offers several important advantages, such as using one modality available at test time to reconstruct the missing modality. In the RGB-D case, this enables the cross-modality scenarios, such as using depth data for semantically segmentation and the RGB images for depth estimation. We demonstrate the effectiveness of the proposed network on two publicly available RGB-D datasets. The experimental results show that the proposed method works well in both semantic segmentation and depth estimation tasks.

1. INTRODUCTION

Visual scene understanding is a critical capability enabling robots to act in their working environment. Modern robots and autonomous vehicles are equipped with many, often complementary sensing technologies. Multiple sensors aim to satisfy the need for the redundancy and robustness critical for achieving the human level of the navigation safety.

The most frequent case is RGB-D cameras collecting color and depth information for different computer vision tasks [5, 15, 28]. As information collected by the depth camera is complementary to RGB images, the depth can help decode structural information of the scene and improve the performance on such tasks as object detection and semantic segmentation [28].

*[1]Giorgio Giannone is with Sapienza University of Rome, Italy, giorgio.c.giannone@gmail.com, this work was done during the internship at Naver Lab Europe.

†[2]Boris Chidlovskii is with Naver Labs Europe, France, boris.chidlovskii@naverlabs.com

Development of convolutional neural networks (CNNs) boosted the performance of the image classification, object detection and semantic segmentation tasks. The key contribution of CNN models lies in their ability to model complex visual scenes. Current CNN-based approaches provide the state-of-the-art performance in semantic segmentation benchmarks [4, 9].

When RGB images are completed with depth information, the straightforward idea is to incorporate depth information into a semantic segmentation framework. Different methods have been developed including deep features pooling, dense feature, multi-scale fusion, etc. [7, 8, 11, 18, 27]. Most recent methods, like FuseNet [13, 16], use an encoder-decoder architecture, where the encoder part is composed of two branches that simultaneously extract features from RGB and depth images and *fuse* depth features into the RGB feature maps. Moreover, training individual RGB and depth models has been replaced with the joint learning. It was shown that the semantics predictions of jointly learned network can be fused more consistently than predictions of a network trained on individual views [23].

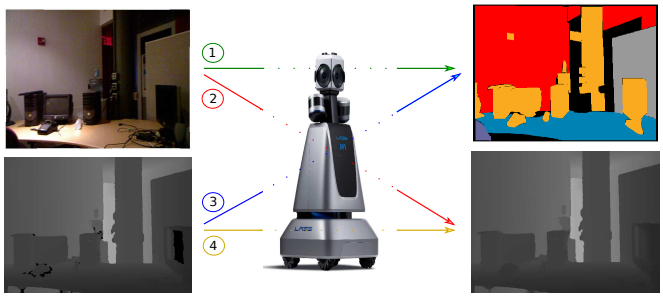

Figure 1. Scenarios for RGB-D data include semantic segmentation from RGB (1), depth (3) or both (1+3), depth prediction from RGB (2) and depth completion from depth (4).

In this paper, we propose a new deep learning architecture for tasks of semantic segmentation and depth estimation from RGB-D images. Usually, these tasks are addressed separately, with a special design for semantic segmentation [13, 23] or depth prediction [7]. We develop a unifying framework capable to cope with either task in different scenarios (see Figure 1).

We adopt the multi-view approach to RGB-D data, where RGB and depth are two channels (modalities) provid-

ing complementary information about a visual scene. All existing methods, whether they train individual models independently or jointly, adopt the fusion-based representation. The feature fusion takes benefit from the view complementarity to reduce the uncertainty of segmentation and labelling. The fusion-based approaches however require both views to be available at train and test time.

We revise the fusion-based approach and replace it with the common representation [24]. Adopting the principle of the common representation gives a number of benefits well known in the multi-view learning [2]. First, it allows to obtain the common representation from one view and then reconstruct all other views. It can accomplish the task when one view is unavailable due to technical or other reasons, thus increasing the robustness and fault-tolerance of the system. Working with one-view data at test time enables *cross-view scenarios* rarely addressed in the state of art. In semantic segmentation, when the RGB view is unavailable, the depth view can be used to obtain the common representation and accomplish the semantic segmentation task. And vice-versa, in the case of a depth estimation, the common representation allows to use the RGB view to reconstruct the depth of a scene or an object.

Second, the common representation is the central component allowing to deploy the same architecture for both RGB-D tasks. Representation common to RGB and depth allows to enforce the consistency between the views and improve the segmentation quality and depth estimation accuracy.

Third, the proposed architecture encourages a higher modularity of the deep network. Our proposal combines the state of art components, the encoder-decoder networks for semantic segmentation and a multi-view autoencoder for the common representation. The system can then benefit from any progress in individual components. The modularity allows to upgrade a component without changing the entire system, training and optimization routines.

The remainder of the paper is organized as follows. In Section 2, we review the state of art of semantic segmentation and depth estimation for RGB-D images. In Section 3, we introduce the multi-view deep architecture and describe in details each component, the two-stage training and optimization. Section 4 reports results of evaluating the network on public NUY2 and SUN datasets; it also discusses some open questions. Section 5 concludes the paper.

2. RELATED WORK

Depth representation. Depth information is rarely used in any segmentation network as raw data, most methods use *HHA representation* of the depth [11]. This representation consists of three channels: disparity, height of the pixels and the angle between normals and the gravity vector based on the estimated ground floor. The color code provided by HHA helps visualize depth information; it can reveal some patterns that resemble RGB patterns.

Semantic segmentation and depth estimation. These two fundamental tasks for RGB-D images are strongly correlated and mutually beneficial, and most efforts were on putting both views in one architecture. In particular, with the success of CNN architectures, many methods aimed to inject the depth information into the semantic segmentation network [8, 13, 16, 18, 23, 29].

Ladicky at al. [18] were first to replace single-view depth estimation and semantic segmentation by a joint training model. They considered both semantic label loss and depth label loss when learning a classifier. Using properties of perspective geometry, they reduced the learning of a pixel-wise depth classifier to a simpler classifier predicting one of fixed canonical depth values [18].

Two separate CNN processing streams, one for each modality, were proposed by Eitel at al. [8]; they are consecutively combined in a late fusion network. The method also introduced a multi-stage training methodology for handling depth data with CNNs. It used the HHA representation of depth and the data augmentation scheme for robust learning with depth images.

A unified framework for joint depth and semantic prediction was proposed by Wang at al. [30]. Given an image, they first use a trained CNN to jointly predict a global layout composed of pixel-wise depth values and semantic labels. The joint network showed to provide more accurate depth prediction than a state-of-the-art CNN trained solely for depth prediction. To further obtain fine-level details, the image is decomposed into local segments for region-level depth and semantic prediction.

By considering RGB and depth channels as multi-modal data, [23] enforced the multi-view consistency during training and testing. At test time, the semantic predictions of the network are fused more consistently than predictions of a network trained on individual views. The network architecture uses a single-view deep learning approach to RGB and depth fusion and enhances it with multi-scale loss minimization.

FuseNet [13] developed an encoder-decoder type of 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.

Although most of the above methods apply the late fusion, it is also possible to fuse depth information into the early layers of fully convolutional neural network [16]. Coupled with the dilated convolution for later contextual reasoning, it combines a depth-sensitive fully-connected CRF with the previous convolution layers to refine the preliminary result.

A step forward from the fusion approach is undertaken

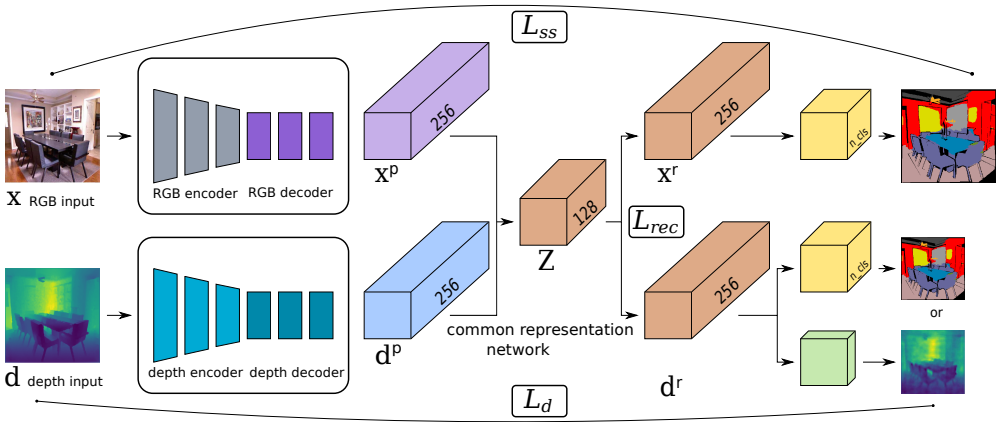

Figure 2. The architecture is composed of two encoder-decoder networks for RGB and depth images and the common representation network. Depending on the setting, the depth network is trained with the segmentation labels or depth ground true values. Better seen in colors.

by Hoffman et al.[14]; they designed a modality hallucination architecture for training an RGB object detection model which incorporates depth information at training time. A convolutional hallucination network learns a new and complementary RGB image representation which is taught to mimic convolutional mid-level features from a depth network. At test time images are processed jointly through the RGB and hallucination networks. Thus, information extracted from depth data is transferred to a network extracting that information from the RGB data.

Depth Completion. The problem of completing the depth channel of an RGB-D image has been addressed in [31]. Indeed, it often the case that commodity-grade depth cameras fail to sense depth for bright, transparent, and distant surfaces thus leaving entire holes in the depth images. They train a deep network that takes a RGB image as input and predicts dense surface normals and occlusion boundaries. Those predictions are then combined with raw depth observations provided by the RGB-D camera to solve for depths for all pixels, including those missing in the original observation.

2.1. Multi-view learning

In the previous section we reviewed different ways to fuse RGB and depth feature maps. Meanwhile, there exist alternative representations for multi-view data [2]. One such alternative, the *common representation learning* (CRL), tries to embed different views of the data in a common subspace [2]. It allows to obtain a common representation from one view and use it to reconstruct other views.

Two complementary approaches to CRL are based on canonical correlation analysis (CCA) and multi-modal autoencoders. CCA based approaches learn a joint representation by maximizing correlation of the views when projected into the common subspace. Second approach to embed two views is based on multi-modal autoencoders (MAEs) [24]. The idea is to train an autoencoder able to perform two kinds of reconstruction. Given one view, the model learns both self-reconstruction and cross-reconstruction (reconstruction of the other view).

As CCA-based and MAE-based approaches appear to be complementary, several methods tried to combine them in one framework [30]. For example, Correlational Neural Network (CorrNet) [3] tried to explicitly maximize the correlation between the views when projecting them into the common subspace. We adopt the idea of CorrNet in our architecture.

3. DEEP ARCHITECTURE

We aim to solve two fundamental tasks for RGB-D images: semantic segmentation and depth prediction. We assume that we are given a training set of N RGB-D images $(\mathbf{x}_i, \mathbf{d}_i)$, $i = 1, \ldots, N$. All $\mathbf{x}_i$ and $\mathbf{d}_i$ images are assumed to be resized to width W and height H. Depth images are in HHA representation and have the same value range as RGB images, $\mathbf{x}_i, \mathbf{d}_i \in R^{H \times W \times 3}$. RGB images are annotated with $\mathbf{y}_i \in L^{H \times W}$, where L is the label set, $L = \{1, \ldots, K\}$. In the case of depth estimation, we assume to have additionally the ground true values $\mathbf{d}_i^* \in R^{H \times W}$.

We propose an architecture composed of two separate branches, one for each modality, which are consecutively fed into a common representation network. Two individual modality networks are of the encoder-decoder type, where the encoder applies dilated convolution to extract an informative feature map, while the decoder applies "atrous" convolution at multiple scales to encode contextual information and refine the segmentation boundaries. This choice is motivated by the recent success of the encoder-decoder architecture of DeepLabV2/V3 networks [4]. It has been also used in FuseNet [13] and SegNet [1] and has showed good

segmentation performance.

Both RGB and depth encoders are initialized by the Resnet101 model pretrained on the MS-COCO dataset. The encoders generate the feature maps, which the decoders use in "atrous" spatial pyramid pooling (ASPP) to robustly segment objects at multiple scales [4].

Feature maps generated by two modality branches are fed into the common representation network implemented in the form of a multi-view autoecoder [30]. Unlike the conventional fusion of RGB and depth feature maps, the multi-view autoencoder allows to extract the shared representation from either one or two views.

3.1. Training RGB network

Our architecture enables two different settings. In the *semantic segmentation* (SS) setting, both RGB and depth views are jointly used to segment an image. In the case of *segmentation and depth estimation* (SS-D), we face the multi-task setting and expect to achieve good results in both tasks.

We first proceed by training two individual modality branches (see the purple and blue sections in Figure 2). Let θ^I and θ^D be parameters of RGB and depth networks, respectively. Let $\mathbf{x}_i^p = g^I(\mathbf{x}_i; \theta^I)$ be the feature map extracted from the last (fully connected) layer of the RGB decoder when applied to image $\mathbf{x}_i$, $\mathbf{x}_i^p \in R^{d \times H^{'} \times W^{'}}$. Analogously, let $\mathbf{d}_i^p = g^D(\mathbf{d}_i; \theta^D)$ be the feature map extracted from depth decoder when applied to the depth image $\mathbf{d}_i$, $\mathbf{d}_i^p \in R^{d \times H^{'} \times W^{'}}$.

The network is trained in two stages. We first train two modality branches, then we train the entire network. In the first stage, we place a randomly initialized softmax classification layer on top of g^I and train the RGB network to minimize the semantic loss of the training data. The semantic loss L_{rgb}^{ss} is defined as the cross-entropy loss

$$L_{rgb}^{ss} = -\sum_{i=1}^{N} P(\mathbf{y}_i | \widehat{\mathbf{x}}_i), \tag{1}$$

where $\widehat{\mathbf{x}}_i$ is a pixel-wise prediction for image $\mathbf{x}_i$, $\mathbf{y}_i$ is the ground truth labels, $P(\mathbf{y}|\mathbf{x}) = \sum_j \log\, p(y_j|x_j)$, and $p(y_j|x_j)$ is the probability of semantic label y_j at pixel j.

The RGB network is trained using the stochastic gradient descent on mini-batches of RGB images. After the convergence, all parameters θ^I of the network are kept for the second stage, except the last layer which will be replaced by the common representation and reconstruction layer.

3.2. Training depth network

Training the depth branch depends on the setting. In the SS setting, the depth network is trained, similarly to the RGB network, to minimize semantic loss $L_d^{ss} = -\sum_{i=1}^{N} P(\mathbf{y}_i | \widehat{\mathbf{d}}_i)$, where $\hat{\mathbf{d}}_i$ is a prediction for depth image $\mathbf{d}_i$.

In the SS-D setting, we train the depth branch to minimize the regression loss on the training depth data. We tested several state-of-art proposals for the loss function L_d^d. One is the scale-invariant loss [7]; it measures the relationships between points in the image irrespectively of the absolute values. We also considered the standard L_2 and Huber loss [10]. Less sensitive to outliers that the L_2 loss, the Huber loss is defined as $L_1^H(\mathbf{d}_i^*, \widehat{\mathbf{d}}_i) = \sum_j D(\mathbf{d}_{ij}^* - \widehat{\mathbf{d}}_{ij})$, where

$$D(x) = \begin{cases} \beta x^2, & \text{if } |x| \leq 1 \\ |x| - \beta, & \text{otherwise.} \end{cases} \tag{2}$$

with $\beta = 0.5$.

3.3. Common representation

Common representation network is implemented as a multi-view autoencoder [3, 24]. It includes a hidden layer and an output layer. The input to the hidden layer is two feature maps $\mathbf{x}^p, \mathbf{d}^p$ fed by two modality branches. Similar to conventional autoencoders, the input and output layer has the same shape as the input, $d \times H' \times W'$, whereas the hidden layer is shaped as $k \times H' \times W'$, with k being often smaller than d (in Figure 2, d=256 and k=128).

Given a two-view input $\mathbf{z} = (\mathbf{x}^p, \mathbf{d}^p)$, the hidden layer computes an encoded representation as the convolution

$$h(\mathbf{z}) = h(\mathbf{W}_x * \mathbf{x}^p + \mathbf{W}_d * \mathbf{d}^p + \mathbf{b}), \tag{3}$$

where $\mathbf{W}_x$, $\mathbf{W}_d$ are projection weights, $\mathbf{b}$ is a bias vector, and h is an activation function, such as *sigmoid* or *tanh*.

The output layer tries to reconstruct $\mathbf{z}$ from this hidden representation $h(\mathbf{z})$ by computing

$$\mathbf{z}^r = g([\mathbf{V}_x * h(\mathbf{z}), \mathbf{V}_d * h(\mathbf{z})] + \mathbf{b}_r), \tag{4}$$

where $\mathbf{V}_x$, $\mathbf{V}_d$ are reconstruction weights, $\mathbf{b}_r$ is a output bias vector, g is an activation function and $[\cdot]$ is the concatenation operation.

Given feature maps $\{(\mathbf{x}_i^p, \mathbf{d}_i^p)\}_{i=1}^N$ from RGB and depth branches, the common representation is designed to minimize the self- and cross-reconstruction errors. The first minimizes the error of reconstructing $\mathbf{x}_i^r$ from $\mathbf{x}_i^p$ and $\mathbf{d}_i^r$ from $\mathbf{d}_i^p$. The second one is the error of reconstructing $\mathbf{x}_i^r$ from $\mathbf{d}_i^p$ and $\mathbf{d}_i^r$ from $\mathbf{x}_i^p$.

To achieve this goal, we try to find the parameter values $\theta^A = \{\mathbf{W}_x, \mathbf{W}_d, \mathbf{V}_x, \mathbf{V}_d, \mathbf{b}, \mathbf{b}_r\}$ by minimizing the reconstruction loss function $L_{rec} = \sum_{i=1}^N L_{rec}^i$, with L_{rec}^i defined on the pair $(\mathbf{x}_i, \mathbf{d}_i)$ as follows

$$l_r(\mathbf{z}_i, g(h(\mathbf{z}_i))) + l_r(\mathbf{z}_i, g(h(\mathbf{x}_i^p))) + l_r(\mathbf{z}_i, g(h(\mathbf{d}_i^p))), \tag{5}$$

where l_r is the reconstruction error, $l_r(\mathbf{x}, \mathbf{x}') = ||\mathbf{x} - \mathbf{x}'||_2^2$. Shorthands $h(\mathbf{x}_i)$ and $h(\mathbf{d}_i)$ denote the representations

$h((\mathbf{x}_i, 0))$ and $h((0, \mathbf{d}_i))$ that are based only on a single view. For each instance with 2 modalities $\mathbf{x}$ and $\mathbf{d}$, $h(\mathbf{x}_i)$ refers to computing the hidden representation using only the $\mathbf{x}$-view. In other words, in Equation (3) for $h(\mathbf{z})$, we set $\mathbf{d}^p = 0$ and obtain $h(\mathbf{z}) = h(\mathbf{W}_x * \mathbf{x}^p + \mathbf{b})$.

In the reconstruction loss L^i_{rec} (5), the first term is the usual autoencoder objective function which helps in learning meaningful hidden representations. The second term ensures that both views can be predicted from the shared representation of the first view alone. The third term ensures that both views can be predicted from the shared representation of the second view alone.

In addition to the common representation and view reconstruction, we also considered a possibility of maximizing the view correlation, as suggested in CorrNet [3]. In such a case, we try to maximize the correlation between the hidden representations of the two views. The correlation term can be included as the fourth term of L_{rec} in (5), it makes sure that the hidden representations of the two views are highly correlated.

3.4. Objective function

Common representation allows to obtain the reconstructed feature maps for both RGB and depth images as $f(g(\mathbf{x}^p))$ and $f(g(\mathbf{d}^p))$. The entire set of network parameters is $\theta = \{\theta^I, \theta^D, \theta^A\}$. The objective function to minimize is then defined as

$$\mathcal{L} = L^{ss}_{rgb} + L_d + \lambda L_{rec}, \tag{6}$$

where the depth branch loss L_d is L^{ss}_d in the SS setting and L^d_d in the SS-D setting; λ is a scaling parameter for the reconstruction loss. In the above formulation, the semantic, depth and reconstruction losses are optimized jointly.

3.5. Training and optimization

The architecture is implemented on the PyTorch framework. At the first stage, we train individual branches independently. In the SS setting, we train RGB and depth branches with segmentation labels, they are denoted RGB-SS and D-SS. Each branch is trained for 20,000 iterations using SGD with momentum 0.9, batch size 24, and minimizing the modality losses, L^{ss}_{rgb} and L^{ss}_d. We retain the model parameters θ^I and θ^D for the second stage.

In the SS-D setting, we train the RGB branch with segmentation labels (RGB-SS) and the depth branch with depth ground truth (D-D). The D-D branch is trained using the scale irrelevant loss, standard L_2 loss or the Huber loss. We apply weight decay 0.0005 and the polynomial decay for the learning rate, with the base LR 0.0001 and power 0.9.

In the second stage, we start with the two branch parameters θ^I and θ^D trained in the first stage, and refine them as well as the common representation network θ^A by minimizing the objective function $\mathcal{L}$ which combines semantic, depth and reconstruction losses. We fine-tune the entire network with the Adam optimizer, but we freeze parameters of two modality encoders, it allows to speed-up the training without performance loss.

For the segmentation task, we additionally perform the data augmentation, by flipping and randomly rotating input images on an angle between [-10, 10] degrees. RGB-D images to be augmented are selected randomly, but the augmentation is identical for both views.

4. EVALUATION

We evaluated the proposed network on two publicly available RGB-D datasets: NYU depth dataset, 2nd version [27] and SUN [28]. NYU2 is a popular dataset, with 27 indoor categories. As not all categories are well represented, the publicly available split [27] reorganized the dataset into 13 most common categories and *other* category for all remaining images. The training/test split is 795/654 images. Images are resized to 512×512 at training time, full size images are used at test time.

SUN dataset contains 10,335 RGB-D images with 40 categories [28]. Following the publicly available split with 37 most common and *other* categories [12], it consists of 5,285 images for training and 5,050 images for testing. Images are resized to 360×360 at training time; full size images are used at test time. All depth images are encoded using the HHA representation.

4.1. Qualitative analysis

We start with the qualitative analysis and test the proposed architecture with exemplar RGB-D images. Figure 3.a shows how a NYU2 example gets processed by the network. In addition to the input images and ground truth segmentation, it shows feature maps extracted at different layers of the network. The upper row refers to the RGB branch, the lower row refers to the depth branch. Column 2 visualizes feature maps generated by two modality decoders. Column 3 shows the common representations obtained from each modality map. A close resemblance of the two maps supports the idea that a common representation which can be obtained from either view. Then, the reconstructed feature maps for both modalities are shown in column 4 and final predictions in column 5.

Figure 3.b shows the cross-view reconstruction, where the RGB image is only available at test time. It starts with feature maps extracted from RGB network and the common representation. Then it shows how the common representation is used for two reconstruction and prediction maps.

4.2. Quantitative results

Modularity. The proposed architecture is designed in a modular way. It does not use any particular techniques

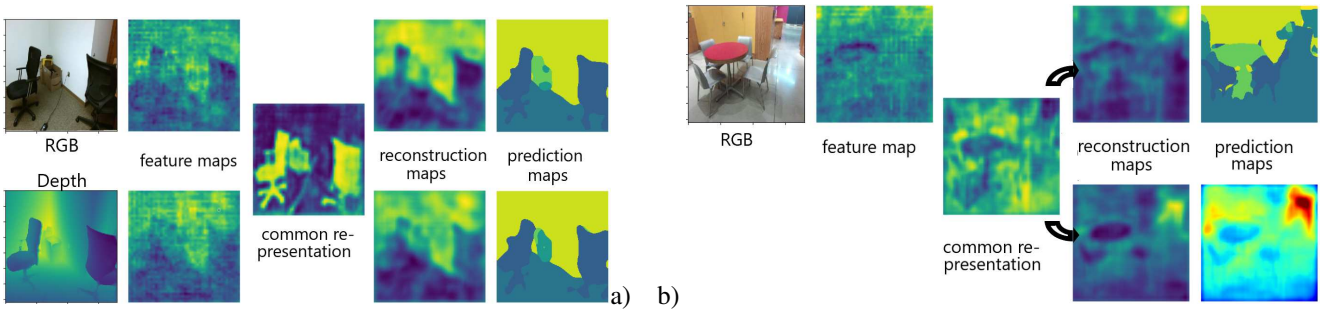

Figure 3. a) Processing RGB-D images at different layers of the architecture; b) Reconstruction from one view. Better seen in colors.

invented to improve the semantic segmentation in special cases and image regions, such as multi-scale input, CRF, overlapping windows and cross-view ambiguities [16, 22]. This choice is motivated by two main reasons. First, we wanted to test the effectiveness of the CRL in isolation, by excluding any impact of the additional improvements and by comparing to the state of art baseline. We paid most attention to the multi-view autoencoder and its capacity to generate common representation and reconstruct RGB and depth views. Second, the architecture design is general enough to cope with two different settings (SS and SS-D) and different modalities. We prefer an ability to work multi-modal and multi-task to an architecture narrowed to processing one particular task. Moreover, since the common representation is complementary to many of the state of art improvements, the proposed architecture can integrate most of them to further boost the performance.

Evaluation metrics. To evaluate our network on the segmentation task, we prefer the intersection-over-union (IoU) score to the pixel accuracy. The pixel accuracy is known for being sensitive to the class disbalance, when many images include the large objects such as bed, wall, floor, etc.. Therefore, the accuracy value may be misleading when the network performs better on the large objects and worse on the small ones. Instead, IoU score remains informative on both balanced and unbalanced datasets.

Let C_{ij} denote the number of pixels those are predicted as class j but actually belongs to class i, where $i, j \in L$. Then C_{ii} denotes the number of pixels with correct prediction of class i. Let T_i denote the total number of pixels that belongs to class i in the ground truth, K is the total number of classes in the dataset. Then IoU is the average value of the intersection between the ground truth and the predictions: $IoU = \frac{1}{K} \sum_i \frac{C_{ii}}{T_i + \sum_j C_{ji} - C_{ii}}$.

For depth estimation, we use the root mean square error (RMSE) that measures the error between the estimated depth and ground truth.

Hyper-parameters. We set $W^{'} = H^{'} = 65$ for the NYU2 set and $W^{'} = H^{'} = 46$ for the SUN set. Feature maps generated by the modality branches are shaped with $d = 256$. The number of hidden variables in the common representation is fixed, $k = 128$. During the training with the objective function $\mathcal{L}$ in (6), weight λ of reconstruction loss is 1.

4.3. Semantic segmentation and depth estimation

We consider three different ways to use the architecture presented in Section 3 to process the RGB-D data.

- **Independent learning**: In this case, each modality branch is trained and tested independently. In the SS setting, we cope with RGB-SS and D-SS branches; in the SS-D setting, we train and test RGB-SS and D-D branches to provide the baseline performances.
- **Joint learning**: The network is trained in two stages as described in Section 3. In SS and SS-D settings, the network is trained to minimize the objective function $\mathcal{L}$, with the corresponding depth branch loss L_d (see Section 3.4). In either case, we compare them to the baselines obtained with the independent training. In the SS setting, we test the common representation with one or two modalities available at test time, where the semantic segmentation is evaluated using the RGB, depth or both images. In the SS-D setting, we test the semantic segmentation and depth estimation with one or two modalities available at test time.

Table 1 reports IoU values for the SS and SS-D settings on NYU2 dataset. In the SS setting, training two modality branches independently yields 53.1 (RGB) and 37.1 (depth) IoU values; this reflects the RGB input being more informative than depth. Learning a joint model and using the common representation at test time improves the performance in cases when depth or both views are available. As both views address the segmentation task, the common representation makes performance dependent on which modality is available at test time. Instead it does not depend which modality is being reconstructed.

In the SS-D setting, the baseline for RGB-SS branch is the same, the baseline for depth reconstruction using D-D branch gives RMSE value of 0.51. The common representation improves the RGB value to 54.3, and reduces the reconstruction error to 0.39 and 0.53 when using the depth

only or both views, respectively. In the cross-view reconstruction, using depth for segmentation drops IoU value to 35.0 only, using RGB for depth estimation yields 0.72 error.

Branch	Independent		Common representation		
	RGB	Depth	RGB	Depth	RGB+D
NYU2 dataset, SS setting					
RGB-SS	53.1	-	54.1	41.2	**57.6**
D-SS	-	37.1	54.2	41.1	57.7
NYU2 dataset, SS-D setting					
RGB-SS	53.1	-	54.3	35.0	**55.2**
D-D	-	0.51	0.72	**0.39**	0.53
SUN dataset, SS setting					
RGB-SS	39.7	-	39.4	31.1	**42.4**
D-SS	-	31.1	39.4	31.1	42.3
SUN dataset, SS-D setting					
RGB-SS	39.7	-	39.3	20.3	**39.9**
D-D	-	0.36	0.62	**0.31**	**0.31**

Table 1. Independent and joint learning with one or two views at test time. The best results are shown in bold.

Methods	Sem. Segmentation			Depth
	RGB	D	RGB+D	RGB
NYU2 dataset				
Our method	54.1	41.2	57.6	0.72
Li et al. [20]	-	-	-	0.82
Roy et al. [25]	-	-	-	0.74
Laina et al. [19]	-	-	-	**0.57**
Eigen et al. [6]	-	-	52.6	0.64
FuseNet-SF3 [13]	-	-	56.0	-
MVCNet [23]	-	-	**59.0**	-
SUN dataset				
Our method	39.49	**31.1**	42.4	0.62
Segnet [1]	22.1	-	-	-
Bayes-Segnet [17]	30.7	-	-	-
Hazirbas [13]	32.4	28.8	33.6	-
FuseNet-SF5 [13]	-	-	37.3	-
DFCN-DCRF [16]	-	-	39.3	-
Context-CRF [26]	42.3	-	-	-
RefineNet [22]	**45.9**	-	-	-
CFN [21]	-	-	**48.1**	-

Table 2. Comparison to the state of art on different tasks.

Table 1 also reports evaluation results on SUN dataset. Using both modalities does improve the performance, moreover depth estimation benefits more from the common representation than the segmentation task.

We compare our results to the state of art on four typical scenarios for RGB-D images (Table 2). Our architecture is the only one able to cope with all the cases. Moreover it remains competitive to the highly specialized architectures [16, 20] which cope with one or two scenarios only.

4.4. Discussion

Both quantitative and qualitative results validated the effectiveness of learning the common representation from RGB and depth images. However the conducted experiments left some questions open; we discuss them in this section.

In addition to the results reported in Tables 1 and 2, we tested a number of alternatives and made some conclusions. First, adding the view correlation term to the reconstruction loss (see Section 3.3) does not seem to improve the common representation nor the performance. Second, the scale-irrelevant loss for the depth estimation, mentioned in Section 3.2, does not seem to perform better than the L_2 and Huber losses; all SS-D results in Tables 1 and 2 refer to the Huber loss.

The two-stage training of the network enables to play with a so-called frozen configuration. The modality branches trained at the first stage get frozen and extract feature maps for all RGB-D images in the dataset. Such a frozen configuration allowed to test different configurations of common representation network before training the full network at the second stage. Below we finally mention some ideas on further improving the current architecture.

1. The common representation is currently limited to one hidden layer. Using deeper multi-view autoencoders has been beneficial in the frozen case.

2. Learning the common representation is implemented on one fixed scale ($k = 128$) of the RGB and depth feature maps. We consider replacing one-fixed-scale MAE with multi-scale ones, on each level of the encoder-decoder networks.

3. ResNet101 model pre-trained on COCO dataset fits well the segmentation task, but to the less extend the depth estimation task. We consider setting up a more appropriate pre-trained model or an option of training it from scratch or combine the two models [16].

5. CONCLUSION

We proposed a new deep learning architecture for the tasks of semantic segmentation and depth prediction from RGB-D images. In the proposed architecture, the conventional feature fusion is replaced with a common deep representation of the RGB and depth views. Combined with an encoder-decoder type of the network, the architecture allows for a joint learning for the semantic segmentation and depth estimation based on their common representation. This approach offers several important advantages, such as using one modality at test time to build a common representation and to reconstruct the missing modality. We reported a number of evaluation results on two standard

RGB-D datasets. Both quantitative and qualitative results validated the effectiveness of learning the common representation from RGB and depth images.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. PAMI*, 39(12):2481–2495, 2017.

[2] T. Baltrusaitis, Ahuja C., and Morency L.-Ph. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. PAMI*, 41(2):423–443, 2019.

[3] S. Chandar, Mitesh M. Khapra, Hugo Larochelle, and Balaraman Ravindran. Correlational neural networks. *Neural computation*, 28(2):257–285, 2016.

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. PAMI.*, 40(4):834–848, 2018.

[5] M. Cordts, Omran M., Ramos S., Rehfeld T., Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *Proc. CVPR*, pages 3213–3223, 2016.

[6] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proc. IEEE ICCV*, pages 2650–2658, 2015.

[7] D. Eigen, Ch. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale seep network.

[8] A. Eitel, J. T. Springenberg, Luciano Spinello, Martin A. Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. *Proc. IEEE/RSJ IROS*, pages 681–687, 2015.

[9] A. Garcia-Garcia, Sergio Orts, Sergiu Oprea, Victor Villena-Martinez, and José García Rodríguez. A review on deep learning techniques applied to semantic segmentation. *CoRR*, abs/1704.06857, 2017.

[10] Ross B. Girshick. Fast R-CNN. In *Proc. IEEE ICCV*, pages 1440–1448, 2015.

[11] S. Gupta, Ross B. Girshick, Pablo Andrés Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proc. ECCV*, pages 345–360, 2014.

[12] A. Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *Proc. CVPR*, pages 4077–4085, 2016.

[13] C. Hazirbas, L. Ma, Csaba Domokos, and Daniel Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Proc. ACCV*, pages 213–228, 2016.

[14] J. Hoffman, S. Gupta, and T. Darrell. Learning with side information through modality hallucination. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 826–834, 2016.

[15] J. Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art. *CoRR*, abs/1704.05519, 2017.

[16] J. Jiang, Zhijun Zhang, Yongqian Huang, and Lunan Zheng. Incorporating depth into both CNN and CRF for indoor semantic segmentation. *CoRR*, abs/1705.07383, 2017.

[17] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian Segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *CoRR*, abs/1511.02680, 2015.

[18] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective.

[19] I. Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proc. IEEE Intern. Conf. on 3D Vision (3DV)*, pages 239–248, 2016.

[20] B. Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proc. CVPR*, pages 1119–1127, 2015.

[21] D. Lin, Guangyong Chen, Daniel Cohen-Or, Pheng-Ann Heng, and Hui Huang. Cascaded feature network for semantic segmentation of rgb-d images. In *Proc. IEEE ICCV*, pages 1320–1328, 2017.

[22] G. Lin, Anton Milan, Chunhua Shen, and Ian D Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proc. CVPR*.

[23] L. Ma, Jörg Stückler, Christian Kerl, and Daniel Cremers. Multi-view deep learning for consistent semantic mapping with rgb-d cameras. *Proc. IEEE/RSJ IROS*, pages 598–605, 2017.

[24] J. Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *Proc. ICML*, pages 689–696, 2011.

[25] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Proc. CVPR*, pages 5506–5514, 2016.

[26] F. Shen, Rui Gan, Shuicheng Yan, and Gang Zeng. Semantic segmentation via structured patch prediction, context crf and guidance crf. In *Proc. IEEE CVPR*, volume 8, pages 5178–5186, 2017.

[27] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proc. ECCV*, pages 746–760, 2012.

[28] S. Song, P. Lichtenberg, and J. Xiao. SUN RGB-D: A RGB-D scene understanding benchmark suite. pages 567–576, 2015.

[29] A. Valada, J. Vertens, A. Dhall, and W. Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *IEEE Intern. Conf. Robotics and Autom. (ICRA)*, pages 4644–4651, 2017.

[30] P. Wang, Xiaohui Shen, Zhe L. Lin, Scott Cohen, Brian L. Price, and Alan L. Yuille. Towards unified depth and semantic prediction from a single image.

[31] Y. Zhang and Th. A. Funkhouser. Deep depth completion of a single RGB-D image. In *Proc. IEEE CVPR*, pages 175–185, 2018.