

# Learning Deep Multimodal Feature Representation with Asymmetric Multi-layer Fusion

Yikai Wang<sup>1\*</sup>, Fuchun Sun<sup>1</sup>, Ming Lu<sup>2</sup>, Anbang Yao<sup>2</sup>

<sup>1</sup>Beijing National Research Center for Information Science and Technology (BNRist),

State Key Lab on Intelligent Technology and Systems,

Department of Computer Science and Technology, Tsinghua University

<sup>2</sup>Cognitive Computing Laboratory, Intel Labs China

{wangyk17@mails., fcsun@}tsinghua.edu.cn, {ming1.lu, anbang.yao}@intel.com

## ABSTRACT

We propose a compact and effective framework to fuse multimodal features at multiple layers in a single network. The framework consists of two innovative fusion schemes. Firstly, unlike existing multimodal methods that necessitate individual encoders for different modalities, we verify that multimodal features can be learnt within a shared single network by merely maintaining modality-specific batch normalization layers in the encoder, which also enables implicit fusion via joint feature representation learning. Secondly, we propose a bidirectional multi-layer fusion scheme, where multimodal features can be exploited progressively. To take advantage of such scheme, we introduce two asymmetric fusion operations including channel shuffle and pixel shift, which learn different fused features with respect to different fusion directions. These two operations are parameter-free and strengthen the multimodal feature interactions across channels as well as enhance the spatial feature discrimination within channels. We conduct extensive experiments on semantic segmentation and image translation tasks, based on three publicly available datasets covering diverse modalities. Results indicate that our proposed framework is general, compact and is superior to state-of-the-art fusion frameworks.

## CCS CONCEPTS

• Computing methodologies → Computer vision tasks; Scene understanding; Computer vision representations.

## KEYWORDS

Multimodal Learning; Compact Network Design; Bidirectional Fusion; Asymmetric Operations

## ACM Reference Format:

Yikai Wang, Fuchun Sun, Ming Lu, Anbang Yao. 2020. Learning Deep Multimodal Feature Representation with Asymmetric Multi-layer Fusion. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, October 12–16, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3394171.3413621>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

MM '20, October 12–16, 2020, Seattle, WA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7988-5/20/10...\$15.00

<https://doi.org/10.1145/3394171.3413621>

## 1 INTRODUCTION

Understanding complex visual scenes is an essential prerequisite for robots and autonomous vehicles to operate in real-world. With the increasing availability of multimodal sensors, fusing information collected by these sensors has shown improved performance on several tasks, like scene recognition, semantic segmentation, etc. In typical multimodal settings, RGB and depth inputs have been widely used to date. Besides, other multimodal inputs such as normals, shadings, textures and edges, are also discussed to some extent. It has been largely verified that the way to effectively fuse multimodal features is essential to the final performance. In this work, we mainly focus on multimodal inputs which are aligned in pixel-level (e.g., RGB, depth and shade), and tackle two drawbacks of existing multimodal fusion works relying on deep neural networks, by proposing two innovative fusion schemes.

Firstly, existing multimodal training methods follow a common design practice that an individual encoder branch is specialized for each modality. For example, regarding to the semantic segmentation task, FuseNet [13], RDFNet [17], SSMA [28] adopt two equal-sized encoders for RGB and depth inputs respectively. The underlying reason may be that for different modalities, different characteristics and feature statistics are not compatible in a single model. However, despite of the heavy parameter load, it prevents the possibility of multimodal features to implicitly fuse during joint training. To tackle this issue, we find that individual encoders are not necessary for multimodal inputs as long as Batch Normalization layers (BNs) [16] are privatized. Specifically, we share all convolutional filters in both encoder and decoder, but adopt modality-specific BNs in the encoder. Modality-specific BNs estimate the channel-wise running mean and variance of activations for each modality separately, and also learn individual channel-wise scale and bias. We empirically verify the effectiveness of this scheme on various modalities, including RGB, depth, normal, shade, etc. On the one hand, this training scheme largely reduces the number of parameters needed for multimodal training. On the other hand, it allows a single network to exploit multimodal features simultaneously, which improves the generalization of convolutional neural networks and achieves better training performance in practice.

Secondly, key ingredients of multimodal fusion include how to design fusion blocks and where to implement fusion. It is verified that exploiting multi-layer fusion, i.e., fusing multimodal features

\* This work was done when Yikai Wang was an intern at Intel Labs China, supervised by Anbang Yao who is responsible for correspondence.

at multiple stages of the network, will improve the fusion performance. In early multimodal fusion works, fusion could be simply realized by feature concatenation, addition or average. Recently, to enable more powerful feature alignment, some multi-layer fusion works adopt a pile of  $3 \times 3$  convolutional layers [17] or attention-based designs [28] for fusion. However, as we will explain, these fusion methods tend to learn symmetric features when followed by a pointwise convolutional layer. This can be simply understood as  $A \rightarrow B$  and  $B \rightarrow A$  fusion leading to the same expressive ability of feature maps (regardless of the order of channels). In this work, we propose a bidirectional fusion scheme, enabling more sufficient multimodal feature fusion. We argue that although existing symmetric fusion methods are suitable for the unidirectional fusion, they are not very compatible with the bidirectional fusion. Besides, along with the emergence of powerful yet complex fusion blocks, increasing amounts of parameters are introduced when fusing multimodal features at multiple layers. To fit the bidirectional fusion scheme, we propose two brand-new asymmetric multimodal fusion operations. Being a fusion in the cross-channel direction, the channel shuffle operation strengthens the multimodal feature interactions across channels, improving the holistic feature representation ability. Being a fusion in the spatial direction within each channel, the pixel shift operation tends to enhance spatial feature discrimination, capturing fine-grained info at object edges, especially for small and thin objects. Both channel shuffle and pixel shift are plug-in and parameter-free operations.

We apply our schemes to two tasks including semantic segmentation and image translation. Our work is verified on three different datasets containing diverse application scenarios ranging from urban city driving scenes to indoor scenes, covering rich modalities including RGB, depth, shade, normal, texture, and edge. Different network architectures containing ResNet [14], Xception65 [7] and U-Net [27] are adopted as backbones. Experimental results prove the effectiveness and generalization of our proposed schemes.

Main contributions of this work can be summarized as follows:

- We verify that multimodal inputs can be fed into a single network with shared parameters and individual BNs for each modality in the encoder, achieving even higher performance than the common practice which uses individual networks.
- We propose two asymmetric parameter-free fusion operations, enabling bidirectional multi-layer fusion from both channel-level and pixel-level perspectives. These operations strengthen the multimodal feature interactions across channels as well as enhance the spatial feature discrimination.
- By merely introducing about 0.1% additional parameters on a given unimodal network, our fusion method is able to outperform state-of-the-art fusion methods on several datasets.

## 2 RELATED WORKS

**Multimodal Fusion.** Methods to exploit multimodal information have been studied for decades [11–13, 20, 29], which allow better understanding of visual scenes compared to learning with unimodal inputs. Prior works usually rely on hand engineered or learned features extracted from each individual modality and combine features together with designed fusion structures. In [29], Markov random fields are explored for indoor segmentation based on RGB and depth

inputs. [11] improves RGB-D recognition performance by making use of the constructed geometric contour from depth data. More recently, with the success of deep convolutional neural networks, a series of multimodal fusion schemes are proposed for end-to-end feature fusion. Regarding to the fusion position, these works can be categorized into single-layer fusion and multi-layer fusion methods. In schemes of single-layer fusion, multimodal features are usually merged into one branch at a particular layer. For example, [9, 10] stack multimodal inputs by channel-wise concatenation and then feed them to the network. [30] designs a transformation layer which fuses multimodal features between the encoder and decoder. [6, 24] explore multimodal feature fusion at the prediction side. However, it has been verified that single-layer fusion methods can not effectively exploit multimodal features, especially for addressing high-resolution predictions [17, 28]. Besides, in [35], it shows that single-layer fusion is sensitive to the noises in multimodal data. Owing to these factors, multi-layer fusion methods become popular, which combine multimodal features at multiple levels, usually at every downsampling stage of a network. Existing multimodal multi-layer fusion schemes can be further classified into two kinds. The first kind is to directly send the fused features to the decoder side. RDFNet [17] adopts multi-layer fusion at four downsampling stages of the ResNet (encoder), iteratively refines the fused features with the similar idea in RefineNet [22], and then sends these fused features to the decoder. SSMA [28] fuses multimodal features at mid-level and high-level with an attention-based mechanism for feature calibration, and as well sends the fused features to the decoder side. However, this kind of fusion methods prevents the encoder to exploit multimodal features. The second kind is to apply fused features to one of the branches in the encoder for in-depth feature exploiting. Typical works can be traced back to FuseNet [13], which trains two individual branches to learn RGB and depth features, where multiple skip connections in the encoder from the depth branch to RGB branch are used for fusion. [35] also adopts two branches for learning RGB-D features respectively, and merges depth features into RGB branch at all downsampling layers in a hierarchical manner. This kind of methods is illustrated in Figure 2 (a), which will be also discussed later. Such scheme is unidirectional and straightforward, still lacking rich feature interactions and thus may be not sufficient for fusion. To tackle this drawback, we design a bidirectional fusion scheme to improve fusion performance.

**Shuffle and Shift.** In group convolutions [7, 15], outputs that correspond to each channel are only related with a portion of input channels. To address this issue, channel shuffle is presented in ShuffleNet [36] to enhance the information flow across different input channel groups, so that each channel is correlated with all groups. Inspired by ShuffleNet, a recent work [25] allocates each frame feature into groups and then aggregates the grouped features via temporal shuffle operation. To date, the proposed shuffle operations are mostly adopted for strengthening correlations among different groups. In this work, we propose to use the channel shuffle operation for multimodal fusion, partially aiming to promote feature interaction among multimodal features, but also to make use of its asymmetric property, which will be analyzed in Section 3.2. There are also some research works that apply shift operations, which are related to our design. Early in [32], stacking pixel shift layers is treated as parameter-free and FLOP-free operations to

improve spatial information communication. The shift design is further improved in [5], where only a few shift operations are needed, instructed by shift operation penalty and quantization-aware shift learning method. Also in [23], a temporal shift module is proposed to shift a portion of channels along the temporal dimension, aiming to boost information exchange among neighboring frames. In this work, we extend the idea of pixel shift to facilitate spatial correlations among multimodal features, and again, we point out that the shift operation can be another asymmetric fusion operation which is compatible with the proposed bidirectional fusion scheme.

### 3 APPROACH

In this section, we propose two multimodal fusion schemes. The first is a parameter-sharing scheme which compresses the model size and enables implicit feature fusion. The second is a bidirectional fusion scheme which enables explicit and sufficient feature fusion.

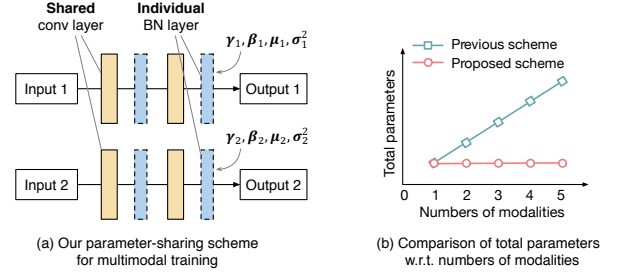
#### 3.1 Parameter-sharing Scheme

As illustrated in Figure 1(a), we provide a multimodal fusion scheme with two input modalities as an example. Unlike existing works which necessitate individual encoders for multiple modalities, we propose that by sharing convolutional parameters but leaving Batch Normalization layers (BNs) [16] modality-specific, we are able to train a single network for multiple modalities. In modern deep neural networks, the mechanism of using BNs has become one of the most successful architectural innovations. A BN layer whitens activations over a mini-batch of features, and transforms the whitened results with channel-wise affine parameters, including scale and bias which provide the possibility of linearly transforming whitened activations to any scales. Sharing network parameters but privatizing BNs has been proved to be effective for efficient model adaption when considering multiple tasks or multiple domains [18, 26, 31, 33]. Inspired by this, we extend the idea of privatizing BNs for multimodal training, where activation statistics of different modalities are normalized separately, and channel-wise scale and bias of BNs are also learned individually for each modality. Network parameters apart from BNs are shared for all modalities. Specifically, assuming there are  $S$  modalities for fusion, for the  $s^{\text{th}}$  modality, where  $s \in \{1, 2, \dots, S\}$ , privatizing BN can be formulated as:

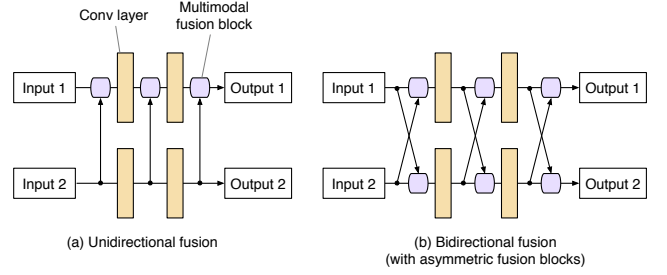
$$\mathbf{y}_s = \gamma_s \cdot \frac{\mathbf{x}_s - \boldsymbol{\mu}_s}{\sqrt{\sigma_s^2 + \epsilon}} + \boldsymbol{\beta}_s, \quad (1)$$

where  $\mathbf{x}_s, \mathbf{y}_s \in \mathbb{R}^{N \times C \times H \times W}$  are the input and output feature maps of the  $s^{\text{th}}$  modality respectively;  $N, C, H, W$  denote the batch size, the number of channels, the height and width of the feature map respectively;  $\boldsymbol{\mu}_s, \sigma_s^2 \in \mathbb{R}^C$  are the mean and standard deviation values of input activations over the current mini-batch of the  $s^{\text{th}}$  modality, calculated by  $\boldsymbol{\mu}_s = \frac{1}{NHW} \sum_{n,h,w} \mathbf{x}_s$  and  $\sigma_s^2 = \frac{1}{NHW} \sum_{n,h,w} (\mathbf{x}_s - \boldsymbol{\mu}_s)^2$ ; besides,  $\gamma_s, \boldsymbol{\beta}_s \in \mathbb{R}^C$  are learnable scale and bias with respect to the  $s^{\text{th}}$  modality;  $\epsilon$  is a small constant to avoid division by zero.

Note that the parameter-sharing indicates sharing all convolutional filters in both encoder and decoder, but privatizing BNs indicates using modality-specific BNs merely in the encoder. We find sharing BNs in the decoder part achieves better results especially for the multimodal image translation task.



**Figure 1:** (a) A compact multimodal fusion scheme, with shared parameters for convolutional layers (also for fully-connected layers, if any) and individual BN parameters. (b) A comparison of total parameters for feature encoding between existing multimodal fusion schemes and ours. With the increasing of total modalities, the size of our scheme is nearly unchanged.

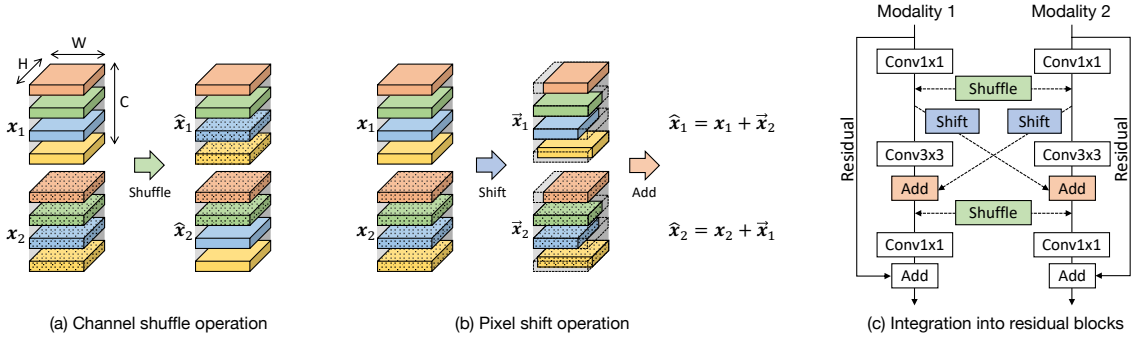


**Figure 2:** (a) Existing multimodal multi-layer fusion schemes mostly adopt unidirectional fusion. (b) Our proposed bidirectional fusion scheme enables each branch to exploit multimodal features. In order to take full advantage of this scheme, we need to design new asymmetric fusion methods.

This training scheme largely compresses model parameters for multimodal fusion. Besides that, learning with shared parameters facilitates interior interactions of multimodal features, which even brings performance improvements. Verification results will be discussed in Table 4, where nearly 50% parameters reduction is obtained with even higher evaluation scores compared with using individual encoders. Such parameter-sharing scheme is not limited to two modalities, which will be verified in Table 3.

#### 3.2 Bidirectional Fusion Scheme

As described in Section 2, early works usually fuse multimodal features at one particular layer, while it has been recently verified that multi-layer fusion (fuse at multiple layers) can exploit supplementary information more adequately. Regarding typical multi-layer fusion works [13, 35], multimodal features are usually fused into one branch and are then further exploited by the encoder. The scheme can be shown as Figure 2(a), where features learned from the second branch are merged into the first branch. This scheme allows the first branch of encoder to exploit fused features. However, as the fusion is unidirectional, the second branch remains unimodal from beginning to end and thus cannot bring informative features at later fusion layers. Besides, two branches would be highly unbalanced during training, and such unbalance may impact the fusion



**Figure 3:** Proposed two asymmetric multimodal fusion operations that are compatible with bidirectional fusion scheme. In the figure, we use a hat sign to represent the fused feature, i.e.,  $\hat{x}_1 = \mathcal{F}(x_1, x_2)$ ,  $\hat{x}_2 = \mathcal{F}(x_2, x_1)$ . Best be viewed in color. (a) Channel shuffle operation exchanges a portion of features with respect to the same-indexed channels. (b) Pixel shift operation performs pixel-wise shifts on four directions, and uses the addition across multimodal features for fusion. (c) We integrate both operations into residual blocks of ResNet architectures for illustration. Blocks in color and dashed lines indicate our inserted structures. Note that the colors of blocks are consistent with the colors of arrows in the first two subfigures, for better understanding.

performance. In our experiments, we find for the unidirectional fusion scheme, different fusion directions lead to very different performance.

From this point of view, it would be worthwhile to design a new kind of multi-layer fusion that can overcome the aforementioned drawback, and improve the fusion performance. To this end, we propose a bidirectional fusion scheme for multi-layer fusion, illustrated in Figure 2(b). In this scheme, multimodal features of different encoder branches are merged mutually, enabling rich feature interactions. However, we find commonly used fusion operations, such as concatenation and average, are not very compatible with the bidirectional fusion scheme. To make it clear, we provide a following definition using two modalities as an example.

**Definition.** Let  $\mathcal{F}(\cdot, \cdot; \theta)$  be a fused feature of two modalities, where  $\theta$  denotes the internal parameters of the fusion block. Let  $C$  be a single pointwise convolutional layer. We define a fusion block as **symmetric** if for any  $\theta_1, C_1$ , there exist  $\theta_2, C_2$  s.t.  $C_1(\mathcal{F}(x_1, x_2; \theta_1)) = C_2(\mathcal{F}(x_2, x_1; \theta_2))$ , holding for any two feature maps  $x_1, x_2$ . We define a fusion block as **asymmetric** if the definition of symmetric fusion does not hold.

The reason for introducing a convolutional layer  $C$  into the definition is because in practice, most fused features are followed by a convolutional layer which further mixes features along the channel. Regarding common fusion methods, it is obvious that addition and average operations are symmetric. The concatenation operation can be proved symmetric when exchanging the order of the same-length outputs for  $x_1$  and  $x_2$ , which can be realized by their following  $C_1$  and  $C_2$ . For these three fusion methods, internal fusion parameters  $\theta = \emptyset$ . Recently proposed fusion methods that apply internal convolutional layers, i.e.,  $\theta \neq \emptyset$ , for example attention-based fusion blocks in SSMA [28], and MMF blocks in RDFNet [17], can also be proved symmetric when  $\theta_2$  is obtained by exchanging two groups of modality-specific parameters of  $\theta_1$ . These commonly used symmetric fusion methods are not very compatible with the bidirectional fusion scheme. See Figure 2(b), during multimodal training, as both final outputs are usually applied with the same

supervision signals, features fused by symmetric fusion methods at both branches tend to learn similar representations (potentially can be the same). This would bring redundant information at both encoder branches.

Based on the above analysis, we propose two kinds of asymmetric fusion operations to fit the bidirectional fusion scheme, called **AsymFusion**. These operations include channel shuffle and pixel shift, which are both parameter-free. We first introduce the designs of both operations, and show how to integrate them into popular network architectures.

**Channel Shuffle.** To strengthen the interaction of multimodal information flow across channels, we propose the channel shuffle operation. As illustrated in Figure 3(a), when given two feature maps  $x_1, x_2 \in \mathbb{R}^{C \times H \times W}$ , channel shuffle fuses two features by exchanging features corresponding to a portion of channels. Given a channel split point  $T$ , satisfying  $1 < T < C, T \in \mathbb{Z}$ , the channel shuffle operation can be described as:

$$\begin{aligned} \mathcal{F}(x_1, x_2) &= x_1[1, \dots, T] \parallel x_2[T+1, \dots, C], \\ \mathcal{F}(x_2, x_1) &= x_2[1, \dots, T] \parallel x_1[T+1, \dots, C], \end{aligned} \quad (2)$$

where  $\parallel$  indicates channel-wise concatenation;  $1, \dots, T$  and  $T+1, \dots, C$  indicate channel indices, which in our experiments, make up 70%, 30% channels respectively. According to this formulation, no additional parameters are introduced, i.e.,  $\theta = \emptyset$ .

An essential property of the shuffle operation leading to its asymmetry is that, two features after shuffle have no feature overlaps. Hence, it is straightforward to find input feature maps  $x_1, x_2$  and  $C_1$ , such that no  $C_2$  is able to result in the same features, which means fusion by channel shuffle is asymmetric.

**Pixel Shift.** To improve spatial information communication of multimodal features, we introduce the second asymmetric fusion operation which uses pixel shift on feature maps, as shown in Figure 3(b). Fusion by pixel shift contains two steps. Supposing the number of channels  $C$  is divisible by 4. For the first step, we divide the feature into four groups, each having  $C/4$  channels, and shift one pixel for every group with one of four spatial directions. This



operation can be formulated as:

$$\begin{aligned}\tilde{x}_1[c, h, w] &= O(x_1)[c, h + \alpha_c + 1, w + \beta_c + 1], \\ \tilde{x}_2[c, h, w] &= O(x_2)[c, h + \alpha_c + 1, w + \beta_c + 1],\end{aligned}\quad (3)$$

where  $O(\cdot)$  indicates a zero padding;  $\alpha_c, \beta_c$  are position indicators,  $\alpha_c = [0, -1, 0, 1]_{\lfloor c/4 \rfloor}$ ,  $\beta_c = [-1, 0, 1, 0]_{\lfloor c/4 \rfloor}$ ; and adding one pixel on position indicators is due to the zero padding.

For the second step, each fused feature is obtained by adding the original feature and the shifted feature of the other modality:

$$\begin{aligned}\mathcal{F}(x_1, x_2) &= x_1 + \tilde{x}_2, \\ \mathcal{F}(x_2, x_1) &= x_2 + \tilde{x}_1,\end{aligned}\quad (4)$$

again, the shift operation brings no additional parameters, i.e.,  $\theta = \emptyset$ . To illustrate that the shift operation is asymmetric, we simply let  $x_1 = 0, x_2 \neq 0$ . For any  $C_1$ , if the definition of symmetric fusion is satisfied, there should always exist  $C_2$  such that  $C_1(\tilde{x}_2) = C_2(x_2)$ . However, this statement cannot be true, as a shifted feature no longer keeps its original pixel alignments across channels.

Both fusion operations introduce no additional parameters, and are also FLOP-efficient. They not only have advantages on asymmetry, but also introduce channel-wise and spatial-wise feature interactions across modalities, respectively, making them more effective for multimodal multi-layer fusion. In Section 4.4, we will show that adopting shift operations in a network may improve predictions for fine-grained objects, and strengthen the representation ability to discriminate rich edge-aware information from context.

Since we have presented two asymmetric fusion operations that can be compatible with the bidirectional fusion scheme, we now consider integrating them into convolutional networks. We adopt residual blocks of ResNet [14] as an example. As shown in Figure 3(c), we insert both fusion operations into the residual blocks of ResNet. As the first  $\text{Conv}1 \times 1$  layer in each residual block performs compression on channel dimension, we speculate that adding shuffle and shift between the two  $\text{Conv}1 \times 1$  layers where features have less channels, will lead to lower information loss during fusion. Specifically, we choose to apply two shuffle operations, for a more sufficient channel fusion, after the first  $\text{Conv}1 \times 1$  layer and before the last  $\text{Conv}1 \times 1$  layer respectively. Besides, we insert the shift operations after the first shuffle. We empirically find that adding the shifted features to features after the  $\text{Conv}3 \times 3$  layers will lead to better performance. Although the pixel shift operation constantly shifts one pixel on a feature map, the corresponding shift of the original image will vary according to the feature map size. To mix different extents of side effects caused by pixel shifts, we apply our fusion approaches at every downsampling stage of the network. Besides fusing at downsampling stages, we also try applying shuffle and shift to other layers but do not observe further improvements.

## 4 EXPERIMENTS

In order to verify the generalization of our proposed schemes, we conduct experiments on two tasks, including semantic segmentation and image translation. The benchmark is performed on three datasets with diverse environments ranging from urban city scenes to indoor scenes, covering a variety of different modalities, including RGB, depth, shade, normal, texture, and edge. Besides, we apply the proposed fusion structures on different network architectures, such as ResNet, Xception65 and U-Net.

### 4.1 Datasets

We consider two indoor datasets NYUDv2 [29], Taskonomy [34], and an outdoor dataset CityScapes [8].

**NYUDv2** is one of the most popular RGB-D datasets in the literature for indoor scene labeling, containing 1449 densely labeled pairs of RGB and depth frames. Following the standard settings, we use 795 training RGB-D pairs and 654 testing RGB-D pairs, and we evaluate our network on 40 classes with labels provided by [11].

**Cityscapes** is a RGB-D dataset for street scene understanding which becomes one of the standard benchmarks. The dataset contains images from 50 different cities, during varying seasons, lighting and weather conditions. The dataset provides 2875 images for training and 500 images for validation. We **do not** use the supplementary training data with coarse annotations.

**Taskonomy** is a large-scale dataset for indoor scene understanding, where each image has other corresponding modalities such as depth, normal, shade, texture, etc., with detailed annotations. We adopt its official sub-dataset containing 9000 samples. As there are no public papers or benchmarks that provide train-test splits to now, we randomly divide the data into 8000 samples for training and the rest 1000 samples for testing, and guarantee no scene overlaps in training and testing samples.

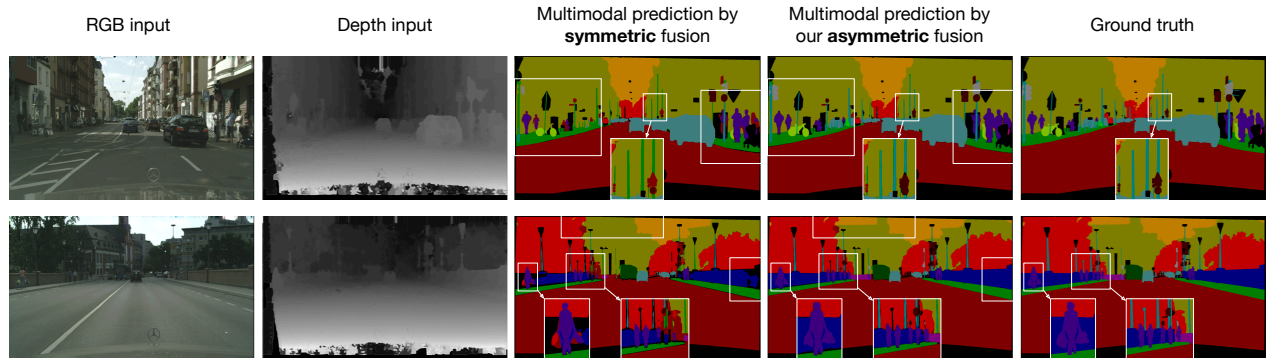
### 4.2 Implementation Details

For semantic segmentation tasks, in order to facilitate comparison with previous fusion approaches, we choose RefineNet [21] for the dataset NYUDv2 and DeepLabv3+ [4] for the dataset Cityscapes, with backbone architectures ResNet [14] and Xception65 [7] respectively. For ResNet architectures, we implement the fusion blocks at all downsampling stages. To make the structure clear, we provide a table showing structure details in supplementary materials. For Xception65, we apply our fusion designs to the last blocks of the entry flow, middle flow and exit flow respectively. As a standard evaluation method, We apply test-time multi-scale evaluation for all experiments and obtain final predications as average results.

For the image translation task, following pix2pix, we adopt a U-Net [27] generator, with an eight-layer encoder and an eight-layer decoder. We adopt multi-layer fusion and apply the fusion with shuffle and shift at the 3<sup>rd</sup>, 5<sup>th</sup>, 7<sup>th</sup> layers of the encoder respectively. We use a discriminator with five convolutional layers for downsampling.

Parameter sharing strategies have been described in Section 3.1. Specifically, for both semantic segmentation and image translation tasks, we share all convolutional parameters and privatize BNs for different modalities in the encoder; we directly share all parameters including BNs in the decoder.

For the semantic segmentation task, we learn an ensemble at the final predictions to further fuse multimodal prediction scores. For  $S$  modalities, the ensemble is learned using a group of importance scores  $\alpha = [\alpha_1 \alpha_2 \cdots \alpha_S]$ , satisfying  $\alpha \geq 0, \sum_{s=1}^S \alpha_s = 1$ , which can be easily implemented with a softmax function. We then adopt the technique of knowledge distillation to force predications of different modalities to mimic the learned ensemble prediction. We find this strategy brings additional improvements for multimodal fusion with negligible extra costs (only  $S$  extra parameters).



**Figure 4:** Illustrative results of semantic segmentation on Cityscapes dataset. In the third column, we choose the attention-based fusion method as the baseline, which achieves state-of-the-art performance as shown in Table 2. We provide results predicted by our asymmetric fusion method in the fourth column. Regions with sharp prediction differences are indicated with white frames. **Best viewed in color at zoom 300%.**

More implementation details, including learning rate and epoch settings, and the method to extend our bidirectional fusion scheme to more modalities, are provided in supplementary materials.

### 4.3 Results

**Semantic Segmentation.** We report results on both NYUDv2 and Cityscapes datasets in Table 1 and Table 2 respectively. Besides commonly used metrics for semantic segmentation, including pixel accuracy, mean accuracy, and intersection over union (IoU), we also compare total parameters of models. In Table 1, we compare our AsymFusion with four state-of-the-art methods. For a quick comparison with RDFNet, which is a RGB-D fusion scheme using RefineNet as its decoder. Using the same encoder ResNet101, total parameters of RDFNet (366.7M) are much larger than RefineNet (118.1M), making it unfriendly for model deployment. In sharp contrast, when given RefineNet (ResNet101) as the unimodal architecture, our fusion model only has 118.2M total parameters, with less than 0.1M additional parameters than RefineNet, and less than 1/3 parameters compared with RDFNet. Similarly, using ResNet152 as encoder, our model merely introduces 0.11% additional parameters compared with RefineNet. Under this extremely compact setting, our fusion models still outperform other methods with a large margin (over 1% absolute gain for IoU). In Table 2, we compare our fusion method to state-of-the-art methods on Cityscapes validation dataset, and our fusion method outperforms previous methods. We do not use the supplementary training set including 20,000 coarse annotations, considering the training costs. Illustrative results shown in Figure 4 verify that our model captures fine-grained details.

**Image Translation.** We benchmark on the dataset Taskonomy for performing image translation tasks, largely due to its data variety. In this part, we consider a wide range of modalities including depth, normal, shade, texture and edge, and aim to translate these data to RGB. Since there are no published methods of multimodal image translation that has such a data variety to date, we implement two commonly used fusion methods (concatenation and average) and two recently proposed fusion methods (attention and MMF). These four methods are treated as baselines for comparison. we

**Table 1:** Semantic segmentation results comparison on NYUDv2 dataset. Baseline methods mostly adopt RefineNet as the unimodal network, except for SCN. For comparison convenience, we also apply our fusion methods on RefineNet, with two backbone architectures ResNet101 and ResNet152 respectively. We report pixel accuracy (%), mean accuracy (%) and mean IoU (%).

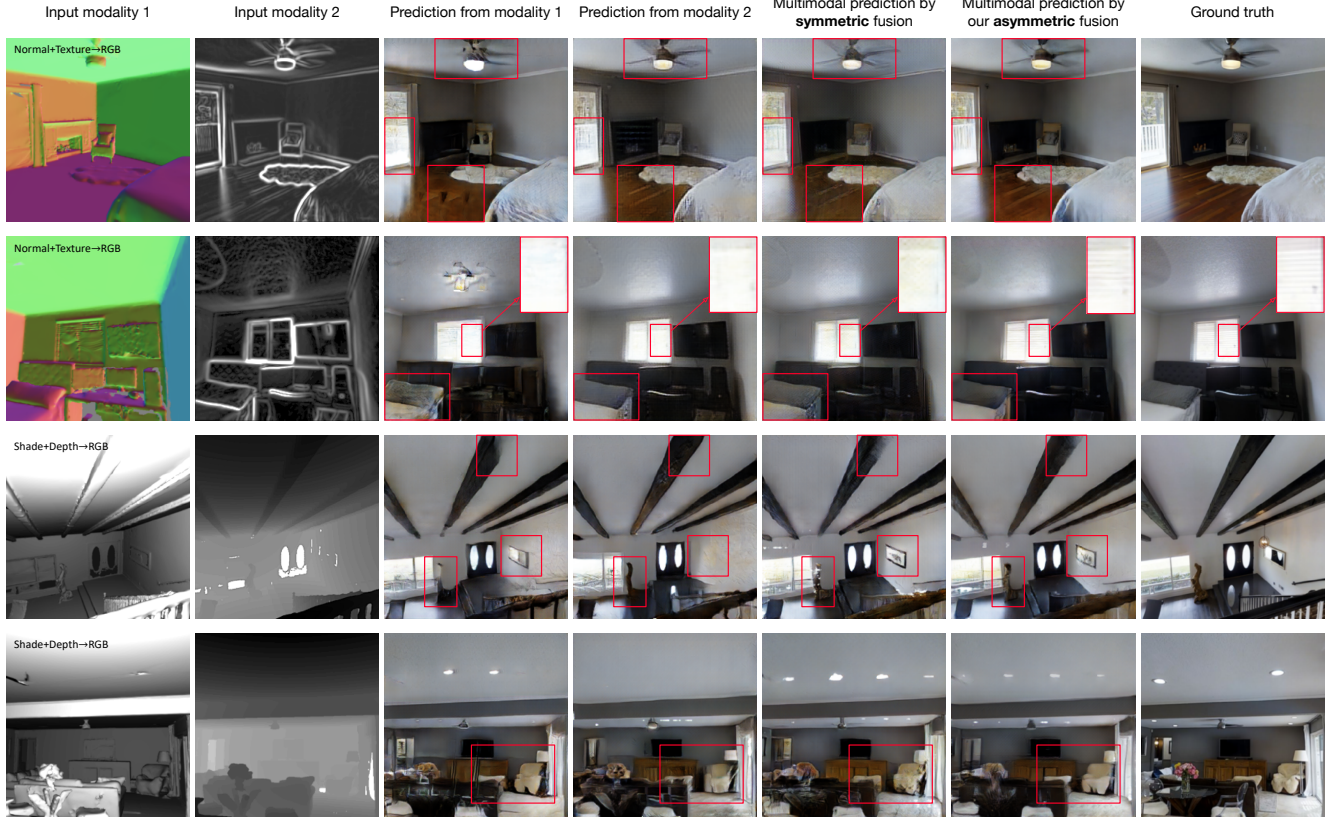
Method	Data modality	Backbone	Pixel acc.	Mean acc.	IoU	#Params.
RefineNet [21]	RGB	ResNet101	73.8	58.8	46.4	118.10M
RefineNet [21]	RGB	ResNet152	74.4	59.6	47.6	133.74M
CFN [19]	RGB-D	ResNet152	-	-	47.7	-
SCN [20]	RGB-D	ResNet152	-	-	49.6	-
RDFNet [17]	RGB-D	ResNet101	75.6	62.2	49.1	366.71M
RDFNet [17]	RGB-D	ResNet152	76.0	62.8	50.1	398.00M
RefineNet †	RGB	ResNet101	73.8	59.0	46.5	118.10M
RefineNet †	Depth	ResNet101	64.0	45.6	34.3	118.10M
<b>AsymFusion</b>	RGB-D	ResNet101	76.6	63.5	50.8	<b>118.20M</b>
<b>AsymFusion</b>	RGB-D	ResNet152	<b>77.0</b>	<b>64.0</b>	<b>51.2</b>	<b>133.89M</b>

† indicates our re-implemented results

adopt Fréchet Inception Distance (FID) score as the evaluation metric. FID compares the statistics of generated images against real images, by fitting a Gaussian distribution to the hidden activations of InceptionNet for compared images and compute Wasserstein-2 distance between these distributions. A lower FID score is better, indicating the generated images are more similar to real counterparts. In Table 3, we provide results comparison for cases with two modalities and three modalities. For different combinations of modalities, our fusion models are consistently better than other methods. Results also indicate that our method can be extended to more modalities and maintain competitive performance. In Figure 5, we provide illustrative examples for comparing different predicted results, where our proposed method outperforms the others.

### 4.4 Ablation Studies

**Importance of Using Private BNs.** A quick question is that, what is the benefit of privatizing BNs? In Table 4, we compare three types of parameter sharing strategies in encoder. We adopt RefineNet (ResNet101), and conduct comparison experiments on NYUDv2 dataset. We obtain two conclusions from the results. Firstly, sharing



**Figure 5:** Image translation with two modalities as inputs, tested on Taskonomy dataset. We provide the result predicted by each single modality respectively, and compare the multimodal fusion performance using symmetric fusion and our asymmetric fusion. For symmetric fusion method, we adopt attention-based method, as it achieves the best performance in Table 3 apart from AsymFusion. The asymmetric method (AsymFusion) contains both shuffle and shift operations. Regions with sharp prediction differences are indicated with red frames.

**Table 2:** Semantic segmentation results on Cityscapes dataset using 19 semantic labels. We apply our fusion methods to DeepLabv3+ (Xception65). We include recently proposed top-performing methods for comparison. Extra data indicates whether use additional training dataset with coarse annotations for further improving performance. We report mean IoU (%) as the evaluation metric.

Method	Data modality	Extra data	Backbone	IoU	#Params.
PSPNet [37]	RGB	×	ResNet101	80.9	56.27M
DeepLabv3 [3]	RGB	×	ResNet101	79.3	58.16M
Mapillary [1]	RGB	×	WideResNet38	78.3	135.86M
DeepLabv3+ [4]	RGB	×	Xception65	78.8	43.48M
DPC [2]	RGB	×	Xception65	80.9	41.82M
DRN [38]	RGB	×	WideResNet38	79.7	129.16M
AdapNet++ [28]	RGB	✓	ResNet50	81.2	30.20M
SSMA [28]	RGB-D	✓	ResNet50	82.2	56.44M
DeepLabv3+ †	RGB	×	Xception65	79.4	43.48M
DeepLabv3+ †	Depth	×	Xception65	62.3	43.48M
<b>AsymFusion</b>	RGB-D	×	Xception65	<b>82.1</b>	<b>43.52M</b>

† indicates our re-implemented results

BNs will lead to an obvious performance drop (4.5% absolute drop for IoU). Secondly, compared with using individual networks, sharing convolutional parameters and leaving BNs privatized brings

**Table 3:** Image translation results comparison on Taskonomy dataset, under different combinations of modalities (two or three modalities) to verify the generalization of our models. All experiments adopt the same fusion layers as described in Section 4.2. FID score is used as the evaluation metric, the lower the better.

Data modality	Concat	Average	Attention	MMF	<b>AsymFusion</b>
Shade,Depth	96.5	101.3	87.3	92.0	<b>82.5</b>
Normal,Texture	88.9	93.0	83.3	85.9	<b>77.8</b>
Depth,Texture,Normal	86.4	90.2	81.5	82.1	<b>75.1</b>
Shade,Normal,Edge	92.8	94.4	85.6	88.6	<b>79.4</b>

even slightly higher performance, yet largely reduces total parameters. Such finding would be instructive for multimodal learning.

**Components Analysis of AsymFusion.** We mainly design AsymFusion using two components, channel shuffle and pixel shift. Besides, we apply knowledge distillation for additional performance improvements, mentioned in Section 4.2. In this part, we verify the importance of these parts. Results shown in Table 5 indicate that shuffle and shift both play importance roles for the fusion performance, which together bring over 3% IoU improvements. The knowledge distillation process further shows slight effects, with about 0.3% IoU improvements.





**Figure 6:** Illustrative semantic segmentation results on Cityscapes dataset, for comparing predictions without and with shift operations. Depth inputs are also used for prediction but are not shown. Red frames indicate regions for prediction, and prediction errors are highlighted in each prediction image. Results show that shift operations may benefit the prediction at small and distant objects.

**Benefits of Using Shift Operations.** In Figure 3, we describe how to integrate pixel shift operations into residual blocks. When integrating shift operations, there are two additional skip connections across two branches. A question arises, is the improvement actually brought by these additional skip connections instead of shift operations? To figure it out, we keep these skip connections and only remove shift operations. The numerical results are 75.5%/62.7%/49.4%, much lower than using shift operations (76.4%/63.1%/50.5%, see Table 5). Illustrative results of predictions without and with shift operations are shown in Figure 6. By comparison, predictions with shift operations tend to be better at capturing details, including thin and small objects, e.g., poles, distant persons and vehicles.

**Comparison of Fusion Directions.** In Table 6, we verify that for multi-layer fusion, the direction of fusion has impact on the fusion performance. For all reported fusion methods, we observe that fusion from depth branch to RGB branch shows better results than fusion with the opposite direction. Applying symmetric fusion methods to the bidirectional scheme shows minor drops than unidirectional counterparts sometimes. We conjecture that symmetric fusion methods tend to have similar feature representation and thus are not very compatible with bidirectional fusion, as described in Section 3.2. Our proposed asymmetric fusion method presents large performance gains when using bidirectional fusion.

## 5 CONCLUSION

We present a compact and effective multimodal fusion framework. To start, we propose that multimodal training can be realized in one single network with modality-specific BNs, enabling implicit fusion via joint feature representation training. Regarding fusion methods, to make advantage of the bidirectional fusion scheme, we propose channel shuffle and pixel shift operations that are asymmetric with

**Table 4:** Exploration of different parameter sharing strategies for the encoder, based on NYUDv2 and Cityscapes datasets. We compare three strategies including using individual networks (widely adopted by existing multimodal works), sharing convolutional parameters and BNs, and sharing only convolutional parameters with individual BNs. We report pixel accuracy (%) and mean IoU (%).

Dataset	Parameter sharing strategy	Pixel acc.	IoU	#Params.
NYUDv2	Individual Convs + Individual BNs	76.1	50.5	236.20M
	Shared Convs + Shared BNs	72.2	46.3	118.10M
	Shared Convs + Individual BNs	<b>76.6</b>	<b>50.8</b>	118.20M
Cityscapes	Individual Convs + Individual BNs	96.8	81.9	87.04M
	Shared Convs + Shared BNs	95.3	78.7	43.48M
	Shared Convs + Individual BNs	<b>97.0</b>	<b>82.1</b>	43.52M

**Table 5:** Comparison different components of our proposed fusion methods, containing channel shuffle, pixel shift, and the knowledge distillation applied at the end of the network. Experiments are conducted with RefineNet (ResNet101) on NYUDv2 dataset. We report pixel accuracy (%), mean accuracy (%) and mean IoU (%).

Shuffle	Shift	Distillation	Pixel acc.	Mean acc.	IoU	#Params.
×	×	×	74.0	58.9	47.3	118.20M
×	×	✓	74.3	59.3	47.6	118.20M
✓	×	×	75.2	62.5	49.3	118.20M
×	✓	×	74.8	61.7	48.7	118.20M
✓	✓	×	76.4	63.1	50.5	118.20M
✓	✓	✓	76.6	63.5	50.8	118.20M

**Table 6:** Results comparison of symmetric fusion methods and the proposed asymmetric fusion method. We compare two settings, i.e., using individual / shared convolutional parameters. Note that individual BNs are adopted for all experiments in this table. Symmetric fusion methods include concatenation, average and attention. The fusion operation addition is omitted as it performs likely to the average operation. We report pixel accuracy (%) / mean IoU (%).

Params.	Fusion direction	Concat	Average	Attention	<b>AsymFusion</b>
Individual	Depth→RGB	74.9 / 48.6	74.7 / 48.2	75.3 / 49.1	75.2 / 49.3
	RGB→Depth	74.0 / 47.1	73.6 / 46.6	74.6 / 48.5	74.9 / 48.7
	Bidirectional	74.7 / 48.0	74.2 / 47.7	74.9 / 48.7	<b>76.2 / 50.5</b>
Shared	Depth→RGB	75.3 / 48.9	74.9 / 48.5	75.5 / 49.4	75.4 / 49.4
	RGB→Depth	74.2 / 47.3	73.9 / 46.8	74.8 / 48.6	75.2 / 49.1
	Bidirectional	75.0 / 48.4	74.6 / 48.2	75.1 / 48.9	<b>76.6 / 50.8</b>

respect to fusion directions. These operations strengthen the interaction of multimodal information flow, and tend to improve feature representation ability to discriminate rich edge-aware information from context. Experimental results on several tasks indicate that our fusion scheme outperforms state-of-the-art counterparts, with only about 0.1% additional parameters than a given unimodal network.

## ACKNOWLEDGEMENT

This work is jointly supported by the National Science Foundation of China (NSFC) and the German Research Foundation (DFG) in project Cross Modal Learning, NSFC 61621136008/DFG TRR-169.

## REFERENCES

- [1] Bulò, S.R., Porzi, L., Kotschieder, P.: In-place activated batchnorm for memory-optimized training of dnns. In: CVPR (2018)
- [2] Chen, L., Collins, M.D., Zhu, Y., Papandreou, G., Zoph, B., Schroff, F., Adam, H., Shlens, J.: Searching for efficient multi-scale architectures for dense image prediction. In: NeurIPS (2018)
- [3] Chen, L., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
- [4] Chen, L., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
- [5] Chen, W., Xie, D., Zhang, Y., Pu, S.: All you need is a few shifts: Designing efficient convolutional neural networks for image classification. In: CVPR (2019)
- [6] Cheng, Y., Cai, R., Li, Z., Zhao, X., Huang, K.: Locality-sensitive deconvolution networks with gated fusion for RGB-D indoor semantic segmentation. In: CVPR (2017)
- [7] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. In: CVPR (2017)
- [8] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016)
- [9] Couprie, C., Farabet, C., Najman, L., LeCun, Y.: Indoor semantic segmentation using depth information. In: ICLR (2013)
- [10] Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: ICCV (2015)
- [11] Gupta, S., Arbelaez, P., Malik, J.: Perceptual organization and recognition of indoor scenes from RGB-D images. In: CVPR (2013)
- [12] Gupta, S., Girshick, R.B., Arbeláez, P.A., Malik, J.: Learning rich features from RGB-D images for object detection and segmentation. In: ECCV (2014)
- [13] Hazirbas, C., Ma, L., Domokos, C., Cremers, D.: Fusetnet: Incorporating depth into semantic segmentation via fusion-based CNN architecture. In: ACCV (2016)
- [14] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
- [15] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
- [16] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
- [17] Lee, S., Park, S., Hong, K.: Rdfnet: RGB-D multi-level residual feature fusion for indoor semantic segmentation. In: ICCV (2017)
- [18] Li, Y., Wang, N., Shi, J., Liu, J., Hou, X.: Revisiting batch normalization for practical domain adaptation. In: ICLR Workshop (2017)
- [19] Lin, D., Chen, G., Cohen-Or, D., Heng, P., Huang, H.: Cascaded feature network for semantic segmentation of RGB-D images. In: ICCV (2017)
- [20] Lin, D., Zhang, R., Ji, Y., Li, P., Huang, H.: SCN: switchable context network for semantic segmentation of RGB-D images. In: IEEE Trans. Cybern. (2020)
- [21] Lin, G., Liu, F., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for dense prediction. In: IEEE Trans. PAMI (2019)
- [22] Lin, G., Milan, A., Shen, C., Reid, I.D.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: CVPR (2017)
- [23] Lin, J., Gan, C., Han, S.: TSM: temporal shift module for efficient video understanding. In: ICCV (2019)
- [24] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
- [25] Ma, P., Zhou, Y., Lu, Y., Zhang, W.: Learning efficient video representation with video shuffle networks. arXiv preprint arXiv:1911.11319 (2019)
- [26] Mudrakarta, P.K., Sandler, M., Zhmoginov, A., Howard, A.G.: K for the price of 1: Parameter-efficient multi-task and transfer learning. In: ICLR (2019)
- [27] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
- [28] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. IJCV (2015)
- [29] Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from RGBD images. In: ECCV (2012)
- [30] Wang, J., Wang, Z., Tao, D., See, S., Wang, G.: Learning common and specific features for RGB-D semantic segmentation with deconvolutional networks. In: ECCV (2016)
- [31] Wang, Y., Sun, F., Li, D., Yao, A.: Resolution switchable networks for runtime efficient image recognition. In: ECCV (2020)
- [32] Wu, B., Wan, A., Yue, X., Jin, P.H., Zhao, S., Golmant, N., Gholaminejad, A., Gonzalez, J., Keutzer, K.: Shift: A zero flop, zero parameter alternative to spatial convolutions. In: CVPR (2018)
- [33] Yu, J., Yang, L., Xu, N., Yang, J., Huang, T.S.: Slimmable neural networks. In: ICLR (2019)
- [34] Zamir, A.R., Sax, A., Shen, W.B., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: CVPR (2018)
- [35] Zeng, J., Tong, Y., Huang, Y., Yan, Q., Sun, W., Chen, J., Wang, Y.: Deep surface normal estimation with hierarchical RGB-D fusion. In: CVPR (2019)
- [36] Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: CVPR (2018)
- [37] Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR (2017)
- [38] Zhuang, Y., Yang, F., Tao, L., Ma, C., Zhang, Z., Li, Y., Jia, H., Xie, X., Gao, W.: Dense relation network: Learning consistent and context-aware representation for semantic image segmentation. In: ICIP (2018)