RefineNet: Multi-Path Refinement Networks
for High-Resolution Semantic Segmentation

Guosheng Lin[1*]   Anton Milan[2]   Chunhua Shen[2,3]   Ian Reid[2,3]

[1]Nanyang Technological University   [2]University of Adelaide   [3]Australian Centre for Robotic Vision

Abstract

Recently, very deep convolutional neural networks (CNNs) have shown outstanding performance in object recognition and have also been the first choice for dense classification problems such as semantic segmentation. However, repeated subsampling operations like pooling or convolution striding in deep CNNs lead to a significant decrease in the initial image resolution. Here, we present RefineNet, a generic multi-path refinement network that explicitly exploits all the information available along the down-sampling process to enable high-resolution prediction using long-range residual connections. In this way, the deeper layers that capture high-level semantic features can be directly refined using fine-grained features from earlier convolutions. The individual components of RefineNet employ residual connections following the identity mapping mindset, which allows for effective end-to-end training. Further, we introduce chained residual pooling, which captures rich background context in an efficient manner. We carry out comprehensive experiments and set new state-of-the-art results on seven public datasets. In particular, we achieve an intersection-over-union score of 83.4 on the challenging PASCAL VOC 2012 dataset, which is the best reported result to date.

1. Introduction

Semantic segmentation is a crucial component in image understanding. The task here is to assign a unique label (or category) to every single pixel in the image, which can be considered as a dense classification problem. The related problem of so-called object parsing can usually be cast as semantic segmentation. Recently, deep learning methods, and in particular convolutional neural networks (CNNs), *e.g.*, VGG [42], Residual Net [24], have shown remarkable results in recognition tasks. However, these approaches exhibit clear limitations when it comes to dense prediction

Figure 1. Example results of our method on the task of object parsing *(left)* and semantic segmentation *(right)*.

in tasks like dense depth or normal estimation [13, 33, 34] and semantic segmentation [36, 5]. Multiple stages of spatial pooling and convolution strides reduce the final image prediction typically by a factor of 32 in each dimension, thereby losing much of the finer image structure.

One way to address this limitation is to learn deconvolutional filters as an up-sampling operation [38, 36] to generate high-resolution feature maps. The deconvolution operations are not able to recover the low-level visual features which are lost after the downsampling operation in the convolution forward stage. Therefore, they are unable to output accurate high-resolution prediction. Low-level visual information is essential for accurate prediction on the boundaries or details. The method DeepLab recently proposed by Chen *et al.* [6] employs atrous (or dilated) convolutions to account for larger receptive fields without downscaling the image. DeepLab is widely applied and represents state-of-the-art performance on semantic segmentation. This strategy, although successful, has at least two limitations. First, it needs to perform convolutions on a large number of detailed (high-resolution) feature maps that usually have high-dimensional features, which are computational expensive. Moreover, a large number of high-dimensional and high-resolution feature maps also require huge GPU memory resources, especially in the training stage. This hampers the computation of high-resolution predictions and usually limits the output size to $1/8$ of the original input. Second, dilated convolutions introduce a coarse sub-sampling of features, which potentially leads to a loss of important details.

Another type of methods exploits features from interme-

diate layers for generating high-resolution prediction, *e.g*., the FCN method in [36] and Hypercolumns in [22]. The intuition behind these works is that features from middle layers are expected to describe mid-level representations for object parts, while retaining spatial information. This information is thought to be complementary to the features from early convolution layers which encode low-level spatial visual information like edges, corners, circles, *etc*., and also complementary to high-level features from deeper layers which encode high-level semantic information, including object- or category-level evidence, but which lack strong spatial information.

We argue that features from all levels are helpful for semantic segmentation. High-level semantic features help the category recognition of image regions, while low-level visual features help to generate sharp, detailed boundaries for high-resolution prediction. How to effectively exploit middle layer features remains an open question and deserves more attentions. To this end, we propose a novel network architecture which effectively exploits multi-level features for generating high-resolution predictions. Our main *contributions* are as follows:

1. We propose a multi-path refinement network (RefineNet) which exploits features at multiple levels of abstraction for high-resolution semantic segmentation. RefineNet refines low-resolution (coarse) semantic features with fine-grained low-level features in a recursive manner to generate high-resolution semantic feature maps. Our model is flexible in that it can be cascaded and modified easily.

2. Our cascaded RefineNets can be effectively trained end-to-end, which is crucial for best prediction performance. More specifically, all components in RefineNet employ residual connections [24] with identity mappings [25], such that gradients can be directly propagated through short-range *and* long-range residual connections allowing for both effective and efficient end-to-end training.

3. We propose a new network component we call *chained residual pooling* which is able to capture background context from a large image region. It does so by efficiently pooling features with multiple window sizes and fusing them together with residual connections and learnable weights.

4. The proposed RefineNet achieves new state-of-the-art performance on 7 public datasets, including PASCAL VOC 2012, PASCAL-Context, NYUDv2, SUN-RGBD, Cityscapes, ADE20K, and the object parsing Person-Parts dataset. In particular, we achieve an IoU score of 83.4 on the PASCAL VOC 2012 dataset, outperforming the currently best approach DeepLab by a large margin.

To facilitate future research, we release both source code and trained models for our RefineNet.[1]

---

[1]Code: https://github.com/guosheng/refinenet

1.1. Related Work

CNNs have become the most successful methods for semantic segmentation in recent years. The early methods in [18, 23] are region-proposal-based methods which classify region proposals to generate segmentation results. Recently, fully convolution networks (FCNNs) [36, 5, 10] have shown effective feature generation and end-to-end training, and have thus become the most popular choice for semantic segmentation. FCNNs have also been widely applied in other dense-prediction tasks, *e.g*., depth estimation [15, 13, 33], image restoration [14], and image super-resolution [12]. The proposed method here is also based on fully convolution-style networks.

FCNN-based methods usually have the limitation of low-resolution prediction. There are a number of proposed techniques which address this limitation and aim to generate high-resolution predictions. The atrous convolution approach DeepLab-CRF in [5] directly output a middle-resolution score map and then applies a dense CRF method [27] to refine boundaries by leveraging color contrast information. CRF-RNN [47] extends this approach by implementing recurrent layers for end-to-end learning of the dense CRF and FCNN. Deconvolution methods [38, 2] learn deconvolution layers to upsample the low-resolution predictions. The depth estimation method in [34] employs super-pixel pooling to output high-resolution prediction.

There exist several methods which exploit middle layer features for segmentation. The FCN method of Long *et al*. [36] adds prediction layers to middle layers to generate prediction scores at multiple resolutions. They average the multi-resolution scores to generate the final prediction mask. Their system is trained in a stage-wise manner rather than end-to-end. The Hypercolumn approach [22] merges features from middle layers and learns dense classification layers. That method also employs a stage-wise training strategy instead of end-to-end training. Both SegNet [2] and U-Net [40] apply skip-connections in the deconvolution architecture to exploit the features from middle layers.

Although several related works exist, it still remains an open question how to effectively exploit middle layer features. We propose a novel network architecture, RefineNet, to address this question. The network architecture of RefineNet is distinct from existing methods. It consists of a number of specially designed components which are able to refine the coarse high-level semantic features by exploiting low-level visual features. In particular, RefineNet employs short-range and long-range residual connections with identity mappings which enable effective end-to-end training of the whole system, and thus help to achieve superior performance. Comprehensive empirical results clearly verify the effectiveness of our novel network architecture for exploiting middle layer features.

2. Background

Before presenting our approach, we first review the structure of fully convolutional networks for semantic segmentation [36] in more detail and also discuss the recent dilated convolution technique [6] which is specifically designed to generate high-resolution predictions.

Very deep CNNs have shown outstanding performance on object recognition problems. Specifically, the recently proposed Residual Network (ResNet) [24] has shown step-change improvements over earlier architectures, and ResNet models pre-trained for ImageNet recognition tasks are publicly available. Because of this, in the following we adopt ResNet as our fundamental building block for semantic segmentation. Note, however, that replacing it with any other deep network is straightforward.

Since semantic segmentation can be cast as a dense classification problem, the ResNet model can be easily modified for this task. This is achieved by replacing the single label prediction layer with a dense prediction layer that outputs the classification confidence for each class at every pixel. This approach is illustrated in Fig. 2(a). As can be seen, during the forward pass in ResNet, the resolution of the feature maps (layer outputs) is decreased, while the feature depth, *i.e.* the number of feature maps per layer (or *channels*) is increased. The former is caused by striding during convolutional and pooling operations.

The ResNet layers can be naturally divided into 4 blocks according to the resolution of the output feature maps, as shown in Fig. 2(a). Typically, the stride is set to 2, thus reducing the feature map resolution to one half when passing from one block to the next. This sequential sub-sampling has two effects: first it increases the receptive field of convolutions at deeper levels, enabling the filters to capture more global and contextual information which is essential for high quality classification; second it is necessary to keep the training efficient and tractable because each layer comprises a large number of filters and therefore produces an output which has a corresponding number of channels. Thus, there is a trade-off between the number of channels and resolution of the feature maps. Typically, the final feature map output ends up being 32 times smaller in each spatial dimension than the original image (but with thousands of channels). This low-resolution feature map loses important visual details captured by early low-level filters, resulting in a rather coarse segmentation map. This issue is a well-known limitation of deep CNN-based segmentation methods.

An alternative approach to avoid lowering the resolution while retaining a large receptive field is to use dilated (atrous) convolution. This method introduced in [6] shows state-of-the-art performance on semantic segmentation. The sub-sampling operations are removed (the stride is changed from 2 to 1), and all convolution layers after the first block use dilated convolution. Such a dilated convolu-

tion, which is effectively a sub-sampled convolution kernel, has the effect of increasing the receptive field size of the filters without increasing the number of weights that must be learned (see illustration in Fig. 2(b)). Even so, there is a significant cost in memory, because unlike the image sub-sampling methods, one must retain very large numbers of feature maps at higher resolutions. For example, if we retain all channels in all layers to be at least 1/4 of the original image resolution, and consider a typical number of filter channels to be 1024, then we can see that the memory capacity of even high-end GPUs is quickly swamped by very deep networks. In practice, therefore, dilation convolution methods usually have a resolution prediction of no more than 1/8 size of the original rather than 1/4, when using a deep network.

In contrast to dilated convolution methods, in this paper we propose a means to enjoy both the memory and computational benefits of deresolving, while still able to produce effective and efficient high-resolution segmentation prediction, as described in the following section.

3. Proposed Method

We propose a new framework that provides multiple paths over which information from different resolutions and via potentially long-range connections is assimilated using a generic building block, the RefineNet. Fig. 2(c) shows one possible arrangement of the building blocks to achieve our goal of high resolution semantic segmentation.

3.1. Multi-Path Refinement

As noted previously, we aim to exploit multi-level features for high-resolution prediction with long-range residual connections. RefineNet provides a generic means to fuse coarse high-level semantic features with finer-grained low-level features to generate high-resolution semantic feature maps. A crucial aspect of the design ensures that the gradient can be effortlessly propagated backwards through the network all the way to early low-level layers over long-range residual connections, ensuring that the entire network can be trained end-to-end.

For our standard multi-path architecture, we divide the pre-trained ResNet (trained with ImageNet) into 4 blocks according to the resolutions of the feature maps, and employ a 4-cascaded architecture with 4 RefineNet units, each of which directly connects to the output of one ResNet block as well as to the preceding RefineNet block in the cascade. Note, however, that such a design is not unique. In fact, our flexible architecture allows for a simple exploration of different variants. For example, a RefineNet block can accept input from multiple ResNet blocks. We will analyse a 2-cascaded version, a single-block approach as well as a 2-scale 7-path architecture later in Sec. 4.3.
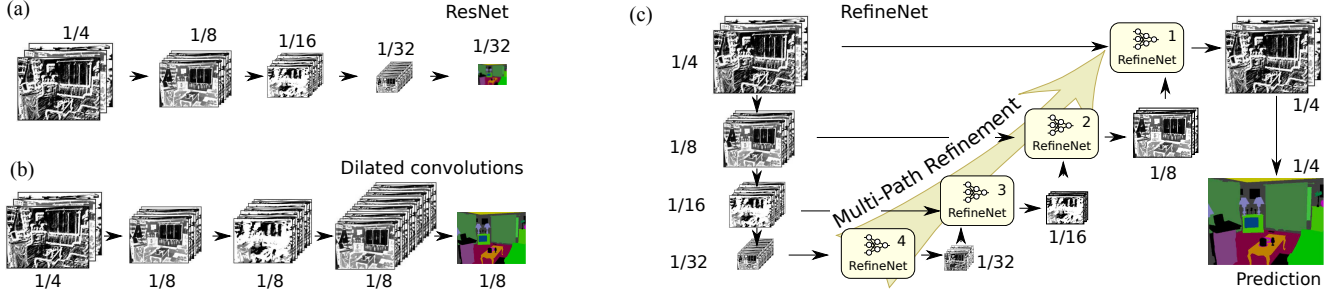
Figure 2. Comparison of fully convolutional approaches for dense classification. Standard multi-layer CNNs, such as ResNet *(a)* suffer from downscaling of the feature maps, thereby losing fine structures along the way. Dilated convolutions *(b)* remedy this shortcoming by introducing atrous filters, but are computationally expensive to train and quickly reach memory limits even on modern GPUs. Our proposed architecture that we call RefineNet *(c)* exploits various levels of detail at different stages of convolutions and fuses them to obtain a high-resolution prediction without the need to maintain large intermediate feature maps. See text and Fig. 3 for details.
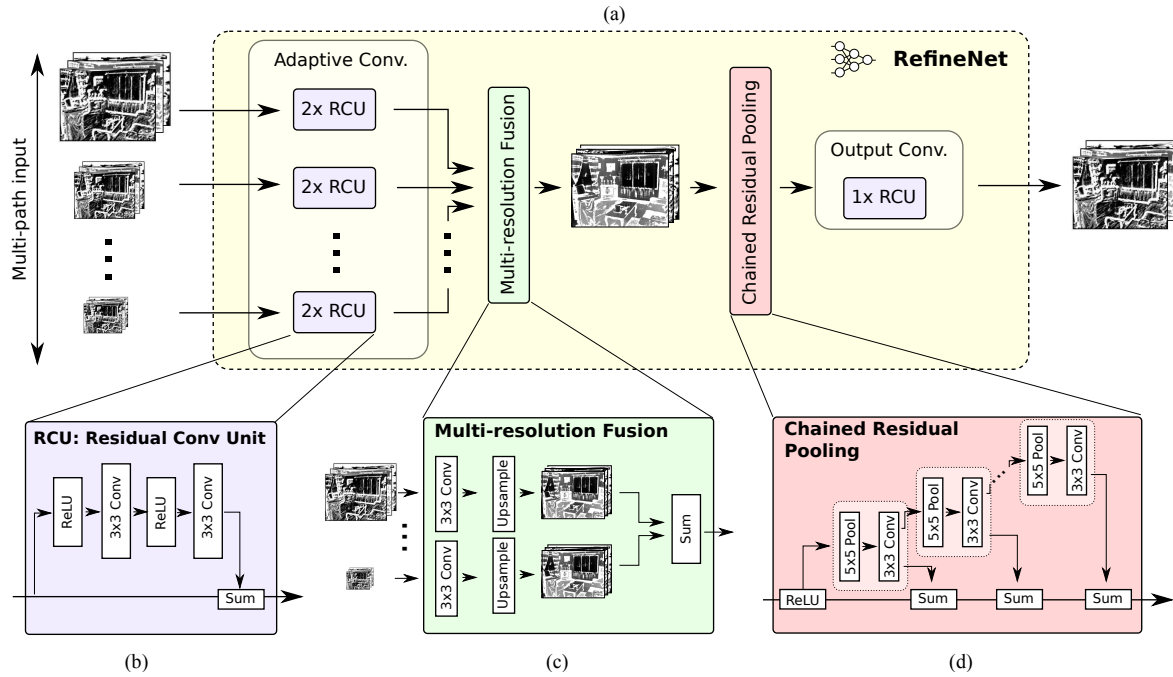

Figure 3. The individual components of our multi-path refinement network architecture RefineNet. Components in RefineNet employ residual connections with identity mappings, allowing gradients to directly propagate both locally through the network and also directly to the input paths via long-range residual connections, and thus enabling effective end-to-end training of the whole system.

We denote RefineNet-m as the RefineNet block that connects to the output of block-m in ResNet. In practice, each ResNet output is passed through one convolutional layer to adapt the dimensionality. Although all RefineNets share the same internal architecture, their parameters are not tied, allowing for a more flexible adaptation for individual levels of detail. Following the illustration in Fig. 2(c) bottom up, we start from the last block in ResNet, and connect the output of ResNet block-4 to RefineNet-4. Here, there is only one input for RefineNet-4, and RefineNet-4 serves as an extra set of convolutions which adapt the pre-trained ResNet weights to the task at hand, in our case, semantic segmentation. In the next stage, the output of RefineNet-4 and the ResNet block-3 are fed to RefineNet-3 as 2-path inputs. The goal of RefineNet-3 is to use the high-resolution features from ResNet block-3 to refine the low-resolution feature map output by RefineNet-4 in the previous stage. Similarly, RefineNet-2 and RefineNet-1 repeat this stage-wise refinement by fusing high-level information from the later layers and high-resolution but low-level features from the earlier ones. As the last step, the final high-resolution feature maps are fed to a dense soft-max layer to make the final prediction in the form of a dense score map. This score map is then upsampled to match the original image using bilinear interpolation. Note that the entire network can be efficiently trained end-to-end.

It is important to note that we introduce long-range residual connections between the blocks in ResNet and the RefineNet modules. During the forward pass, these long-range residual connections convey the low-level features that encode visual details for refining the coarse high-level feature maps. In the training step, the long-range residual connections allow direct gradient propagation to early convolution layers, which helps effective end-to-end training.

3.2. RefineNet

The architecture of one RefineNet block is illustrated in Fig. 3(a). In the multi-path overview shown in Fig 2(c), RefineNet-1 has one input path, while all other RefineNet blocks have two inputs. Note, however, that our architecture is generic and each Refine block can be easily modified to accept an arbitrary number of feature maps with arbitrary resolutions and depths.

Residual convolution unit. The first part of each RefineNet block consists of an adaptive convolution set that mainly fine-tunes the pretrained ResNet weights for our task. To that end, each input path is passed sequentially through two residual convolution units (RCU), which is a simplified version of the convolution unit in the original ResNet [24], where the batch-normalization layers are removed (*cf*. Fig. 3(b)). The filter number for each input path is set to 512 for RefineNet-4 and 256 for the remaining ones in our experiments.

Multi-resolution fusion. All path inputs are then fused into a high-resolution feature map by the multi-resolution fusion block, depicted in Fig. 3(c). This block first applies convolutions for input adaptation, which generate feature maps of the same feature dimension (the smallest one among the inputs), and then upsamples all (smaller) feature maps to the largest resolution of the inputs. Finally, all features maps are fused by summation. The input adaptation in this block also helps to re-scale the feature values appropriately along different paths, which is important for the subsequent sum-fusion. If there is only one input path (e.g., the case of RefineNet-4 in Fig. 2(c)), the input path will directly go through this block without changes.

Chained residual pooling. The output feature map then goes through the chained residual pooling block, schematically depicted in Fig. 3(d). The proposed chained residual pooling aims to capture background context from a large image region. It is able to efficiently pool features with multiple window sizes and fuse them together using learnable weights. In particular, this component is built as a chain of multiple pooling blocks, each consisting of one max-pooling layer and one convolution layer. One pooling block takes the output of the previous pooling block as input. Therefore, the current pooling block is able to re-use the result from the previous pooling operation and thus ac-

cess the features from a large region without using a large pooling window. If not further specified, we use two pooling blocks each with stride 1 in our experiments.

The output feature maps of all pooling blocks are fused together with the input feature map through summation of residual connections. Note that our choice to employ residual connections also persists in this building block, which once again facilitates gradient propagation during training. In one pooling block, each pooling operation is followed by convolutions which serve as a weighting layer for the summation fusion. It is expected that this convolution layer will learn to accommodate the importance of the pooling block during the training process.

Output convolutions. The final step of each RefineNet block is another residual convolution unit (RCU). This results in a sequence of three RCUs between each block. To reflect this behavior in the last RefineNet-1 block, we place two additional RCUs before the final softmax prediction step. The goal here is to employ non-linearity operations on the multi-path fused feature maps to generate features for further processing or for final prediction. The feature dimension remains the same after going through this block.

3.3. Identity Mappings in RefineNet

Note that all convolutional components of the RefineNet have been carefully constructed inspired by the idea behind residual connections and follow the rule of identity mapping [25]. This enables effective backward propagation of the gradient through RefineNet and facilitates end-to-end learning of cascaded multi-path refinement networks.

Employing residual connections with identity mappings allows the gradient to be directly propagated from one block to any other blocks, as was recently shown by He *et al.* [25]. This concept encourages to maintain a clean information path for shortcut connections, so that these connections are not "blocked" by any non-linear layers or components. Instead, non-linear operations are placed on branches of the main information path. We follow this guideline for developing the individual components in RefineNet, including all convolution units. It is this particular strategy that allows the multi-cascaded RefineNet to be trained effectively. Note that we include one non-linear activation layer (ReLU) in the chained residual pooling block. We observed that this ReLU is important for the effectiveness of subsequent pooling operations and it also makes the model less sensitive to changes in the learning rate. We observed that one single ReLU in each RefineNet block does not noticeably reduce the effectiveness of gradient flow.

We have both short-range and long-range residual connections in RefineNet. Short-range residual connections refer to local shortcut connections in one RCU or the residual pooling component, while long-range residual connections refer to the connections between RefineNet modules and

Table 1. Object parsing results on the Person-Part dataset. Our method achieves the best performance (bold).

Method	IoU	Method	IoU
Attention [7]	56.4	HAZN [45]	57.5
LG-LSTM [29]	58.0	Graph-LSTM [28]	60.2
DeepLab [5]	62.8	DeepLab-v2 (Res101) [6]	64.9
RefineNet-Res101 (ours)	**68.6**		

Table 2. Ablation experiments on NYUDv2 and Person-Part.

Initialization	Chained pool.	Msc Eva	NYUDv2	Person-Parts
ResNet-50	no	no	40.4	64.1
ResNet-50	yes	no	42.5	65.7
ResNet-50	yes	yes	43.8	67.1
ResNet-101	yes	no	43.6	67.6
ResNet-101	yes	yes	44.7	68.6
ResNet-152	yes	yes	**46.5**	**68.8**

Table 3. Segmentation results on NYUDv2 (40 classes).

method	training data	pixel acc.	mean acc.	IoU
Gupta et al. [20]	RGB-D	60.3	-	28.6
FCN-32s [36]	RGB	60.0	42.2	29.2
FCN-HHA [36]	RGB-D	65.4	46.1	34.0
Context [30]	RGB	70.0	53.6	40.6
RefineNet-Res152	RGB	**73.6**	**58.9**	**46.5**

the ResNet blocks. With long-range residual connections, the gradient can be directly propagated to early convolution layers in ResNet and thus enables end-to-end training of all network components.

The fusion block fuses the information of multiple shortcut paths, which can be considered as performing summation fusion of multiple residual connections with necessary dimension or resolution adaptation. In this aspect, the role of the multi-resolution fusion block here is analogous to the role of the "summation" fusion in a conventional residual convolution unit in ResNet. There are certain layers in RefineNet, and in particular within the fusion block, that perform linear feature transformation operations, like linear feature dimension reduction or bilinear upsampling. These layers are placed on the shortcut paths, which is similar to the case in ResNet [24]. As in ResNet, when a shortcut connection crosses two blocks, it will include a convolution layer in the shortcut path for linear feature dimension adaptation, which ensures that the feature dimension matches the subsequent summation in the next block. Since only linear transformation are employed in these layers, gradients still can be propagated through these layers effectively.

4. Experiments

To show the effectiveness of our approach, we carry out comprehensive experiments on seven public datasets, which include six popular datasets for semantic segmentation on indoors and outdoors scenes, and one dataset for object parsing called Person-Part. The segmentation quality is measured by the intersection-over-union (IoU) score [16], the pixel accuracy and the mean accuracy [36] over

(a) Test Image   **(b)** Ground Truth   **(c)** Prediction
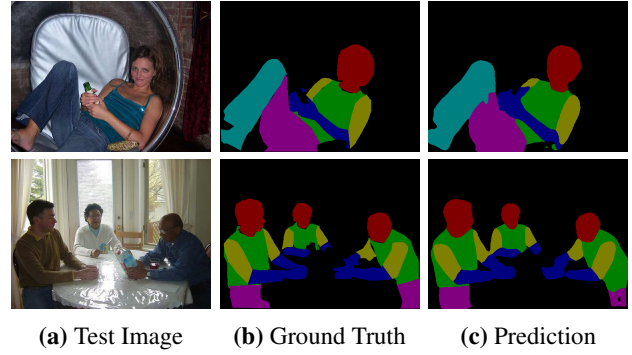
Figure 4. Our prediction examples on Person-Parts dataset.

all classes. As commonly done in the literature, we apply simple data augmentation during training. Specifically, we perform random scaling (ranging from 0.7 to 1.3), random cropping and horizontal flipping of the images. If not further specified, we apply test-time multi-scale evaluation, which is a common practice in segmentation methods [10, 6]. For multi-scale evaluation, we average the predictions on the same image across different scales for the final prediction. We also present an ablation experiment to inspect the impact of components and several variants of our model. Our system is built on MatConvNet [44].

4.1. Object Parsing

We first present our results on the task of object parsing, which consists of recognizing and segmenting object parts. We carry out experiments on the Person-Part dataset [8, 7] which provides pixel-level labels for six person parts including Head, Torso, Upper/Lower Arms and Upper/Lower Legs. The rest of each image is considered background. There are 1717 training and 1818 test images. We use 4 chained pooling blocks for this dataset.

We compare our results to a number of state-of-the-art methods, listed in Table 1. The results clearly demonstrate the improvement over previous works. In particular, we significantly outperform the the recent DeepLab-v2 approach [6] which is based on dilated convolutions for high-resolution segmentation, using the same ResNet as initialization. In Table 2, we present an ablation experiment to quantify the influence of the following components: Network depth, chained residual pooling and multi-scale evaluation (Msc Eva), as described earlier. This experiment shows that each of these three factors can improve the overall performance. Qualitative examples of our object parsing on this dataset are shown in Fig.4.

4.2. Semantic Segmentation

We now describe our experiments on dense semantic labeling on six public benchmarks and show that our RefineNet outperforms previous methods on all datasets.

Table 4. Results on the PASCAL VOC 2012 test set (IoU scores). Our RefineNet achieves the best overall performance (IoU 83.4).

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	potted	sheep	sofa	train	tv	mean
FCN-8s [36]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeconvNet [38]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
CRF-RNN [47]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup [10]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
DPN [35]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Context [30]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
DeepLab [5]	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1	72.7
DeepLab2-Res101 [6]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
CSupelec-Res101 [4]	92.9	61.2	91.0	66.3	77.7	**95.3**	**88.9**	92.4	33.8	88.4	69.1	89.8	**92.9**	87.7	87.5	62.6	89.9	59.2	87.1	74.2	80.2
RefineNet-Res101	**94.9**	60.2	92.8	**77.5**	81.5	95.0	87.4	93.3	39.6	89.3	73.0	**92.7**	92.4	85.4	**88.3**	69.7	92.2	65.3	84.2	**78.7**	82.4
RefineNet-Res152	94.7	**64.3**	**94.9**	74.9	**82.9**	95.1	88.5	**94.7**	45.5	**91.4**	**76.3**	90.6	91.8	**88.1**	88.0	**69.9**	**92.3**	**65.9**	**88.7**	76.8	**83.4**

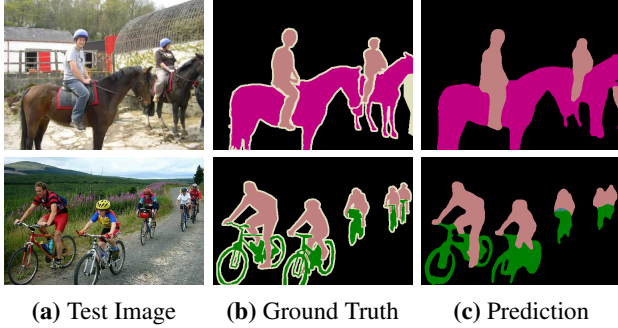**(a)** Test Image   **(b)** Ground Truth   **(c)** Prediction

Figure 5. Segmentation examples on PASCAL VOC 2012.

NYUDv2. The NYUDv2 dataset [41] consists of 1449 RGB-D images showing interior scenes. We use the segmentation labels provided in [19], in which all labels are mapped to 40 classes. We use the standard training/test split with 795 and 654 images, respectively. We train our models only on RGB images without using the depth information. Quantitative results are shown in Tab. 3. Our RefineNet achieves state-of-the-art results on NYUDv2.

Similar to the object parsing task above, we also perform ablation experiments on the NYUDv2 dataset to evaluate the effect of different settings. The results are presented in Tab. 2. Once again, this study demonstrates the benefits of adding the proposed chained residual pooling component and deeper networks, both of which consistently improve the performance as measured by IoU.

PASCAL VOC 2012 [16] is a well-known segmentation dataset which includes 20 object categories and one background class. This dataset is split into a training set, a validation set and a test set, with 1464, 1449 and 1456 images each. Since the test set labels are not publicly available, all reported results have been obtained from the VOC evaluation server. Following the common convention [5, 6, 47, 35], the training set is augmented by additional annotated VOC images provided in [21] as well as with the training data from the MS COCO dataset [31]. We compare our RefineNet on the PASCAL VOC 2012 test set with a number of competitive methods, showing superior performance. We use 4 pooling blocks for this dataset. We also use a dense CRF [27] for further refinement for this

Table 5. Segmentation results on the Cityscapes *test* set.

Method	IoU	Method	IoU
FCN-8s [36]	65.3	DPN [35]	66.8
Dilation10 [46]	67.1	Context [30]	71.6
DeepLab [5]	63.1	DeepLab-v2(Res101) [6]	70.4
LRR-4x [17]	71.8	**RefineNet-Res101 (ours)**	**73.6**

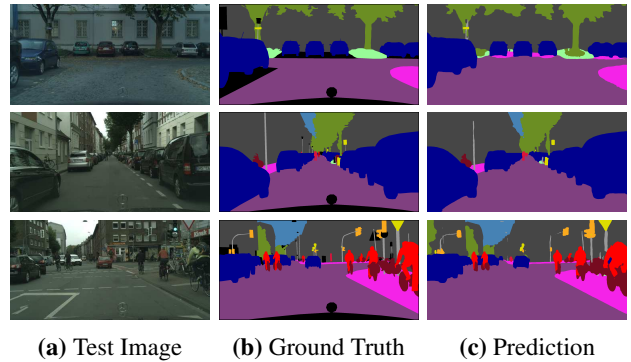**(a)** Test Image   **(b)** Ground Truth   **(c)** Prediction

Figure 6. Our prediction examples on the Cityscapes dataset.

dataset, which yields a marginal improvement of 0.1% on the validation set. Since this dense CRF post-processing only brings very minor improvement on our high-resolution prediction, we do not apply it on other datasets.

The detailed results for each category and the mean IoU scores are shown in Tab. 4. We achieve an IoU score of 83.4, which is the best reported result on this challenging dataset to date.[2] We outperform competing methods in almost all categories. In particular, we significantly outperform DeepLab-v2 [6], which uses the same ResNet-101 network as initialization by nearly 3 percentage points. Selected prediction examples are shown in Fig. 5.

Cityscapes [9] is a very recent dataset on street scene images from 50 different European cities. This dataset provides fine-grained pixel-level annotations of roads, cars, pedestrians, bicycles, sky, *etc*. The provided training set has 2975 images and the validation set has 500 images. In total, 19 classes are considered for training and evaluation. The test set ground-truth is withheld by the organizers, and

---

[2] The result link to the VOC evaluation server: http://host.robots.ox.ac.uk:8080/anonymous/B3XPSK.html

Table 6. Segmentation results on PASCAL-Context dataset (60 classes). DeepLab-v2 uses extra training data from COCO (∼100K images), while we only use the VOC training images.

Method	IoU	Method	IoU
O2P [3]	18.1	CFM [11]	34.4
FCN-8s [36]	35.1	BoxSup [10]	40.5
HO-CRF [1]	41.3	Context [30]	43.3
DeepLab-v2(Res101) [6]	45.7	–	–
RefineNet-Res101 (ours)	47.1	**RefineNet-Res152 (ours)**	**47.3**

Table 7. Segmentation results on SUN-RGBD dataset (37 classes). Our RefineNet significantly outperforms the existing methods.

Method	Train data	Pixel acc.	Mean acc.	IoU
Liu et al. [32]	RGB-D	–	10.0	–
Ren et al. [39]	RGB-D	–	36.3	–
Kendall et al. [26]	RGB	71.2	45.9	30.7
Context [30]	RGB	78.4	53.4	42.3
RefineNet-Res101	RGB	80.4	57.8	45.7
RefineNet-Res152	RGB	**80.6**	**58.5**	**45.9**

we evaluate our method on the their evaluation server. The test results are shown in Table 5. In this challenging setting, our architecture again outperforms previous methods. A few test images along with ground truth and our predicted semantic maps are shown in Fig. 6.

PASCAL-Context. The PASCAL-Context [37] dataset provides the segmentation labels of the whole scene for the PASCAL VOC images. We use the segmentation labels which contain 60 classes (59 object categories plus background) for evaluation as well as the provided training/test splits. The training set contains 4998 images and the test set has 5105 images. Results are shown in Table 6. Even without additional training data and with the same underlying ResNet architecture with 101 layers, we outperform the previous state-of-the-art achieved by DeepLab.

SUN-RGBD [43] is a segmentation dataset that contains around 10,000 RGB-D indoor images and provides pixel labeling masks for 37 classes. Results are shown in Tab. 7. Our method outperforms all existing methods by a large margin across all evaluation metrics, even though we do not make use of the depth information for training.

ADE20K MIT [48] is a newly released dataset for scene parsing which provides dense labels of 150 classes on more than 20K scene images. The categories include a large variety of objects (*e.g.*, person, car, *etc.*) and stuff (*e.g.*, sky, road, *etc.*). The provided validation set consisting of 2000 images is used for quantitative evaluation. Results are shown in Table 8. Our method clearly outperforms the baseline methods described in [48].

4.3. Variants of cascaded RefineNet

As discussed earlier, our RefineNet is flexible in that it can be cascaded in various manners for generating various architectures. Here, we discuss several variants of

Table 8. Segmentation results on the ADE20K dataset (150 classes) *val* set. our method achieves the best performance.

Method	IoU	Method	IoU
FCN-8s [36]	29.4	SegNet [2]	21.6
DilatedNet [5, 46]	32.3	Cascaded-SegNet [48]	27.5
Cascaded-DilatedNet [48]	34.9	–	–
RefineNet-Res101 (ours)	40.2	**RefineNet-Res152 (ours)**	**40.7**

Table 9. Evaluation of 4 types of cascaded RefineNet on NYUDv2.

Variant	Initialization	Msc Eva	IoU
single	ResNet-50	no	40.3
2-cascaded	ResNet-50	no	40.9
4-cascaded	ResNet-50	no	42.5
4-cascaded 2-scale	ResNet-50	no	43.1

our RefineNet, as listed in Tab. 9. The single RefineNet model consists of only one single RefineNet block, which takes all four inputs from the four blocks of ResNet and fuses all-resolution feature maps in a single process. The 2-cascaded version is similar our main model (4-cascaded) from Fig. 2(c), but employs only two RefineNet modules instead of four. For the 2-scale model, the input image is scaled to a factor of 1.2 and 0.6 and fed into 2 independent 4-cascaded ResNets. The detailed architectures of these variants are described in the supplementary document.[3]

The evaluation in Tab. 9 demonstrates that the 4-cascaded version yields better performance than the 2-cascaded and 1-cascaded version, and using 2-scale image input with 2 ResNet is better than using 1-scale input. This is expected due to the larger capacity of the network. However, it also results in longer training time. Hence, we resort to using the single-scale 4-cascaded version as the standard architecture in all our experiments.

5. Conclusion

We have presented RefineNet, a novel multi-path refinement network for semantic segmentation and object parsing. The cascaded architecture is able to effectively combine high-level semantics and low-level features to produce high-resolution segmentation maps. Our design choices are inspired by the idea of identity mapping which facilitates gradient propagation across long-range connections and thus enables effective end-to-end learning. Experiments show that our method sets a new mark for the state of the art in semantic labeling.

---

[3]https://arxiv.org/abs/1611.06612

References

[1] A. Arnab, S. Jayasumana, S. Zheng, and P. H. Torr. Higher order conditional random fields in deep neural networks. In *European Conference on Computer Vision*. Springer, 2016. 8

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, 2015. 2, 8

[3] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *Proceedings of the European Conference on Computer Vision*, 2012. 8

[4] S. Chandra and I. Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *Proceedings of the European Conference on Computer Vision*, 2016. 7

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proceedings of the International Conference on Learning Representations*, 2015. 1, 2, 6, 7, 8

[6] L. 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. *CoRR*, abs/1606.00915, 2016. 1, 3, 6, 7, 8

[7] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. *arXiv preprint arXiv:1511.03339*, 2015. 6

[8] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1971–1978, 2014. 6

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 7

[10] J. Dai, K. He, and J. Sun. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2, 6, 7, 8

[11] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 8

[12] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proceedings of the European Conference on Computer Vision*, 2014. 2

[13] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 1, 2

[14] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE International Conference on Computer Vision*, 2013. 2

[15] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Proceedings of Neural Information Processing Systems*, 2014. 2

[16] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. In *International Journal of Computer Vision*, 2010. 6, 7

[17] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, 2016. 7

[18] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[19] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 7

[20] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from RGB-D images for object detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, 2014. 6

[21] B. Hariharan, P. Arbelaez, L. D. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *Proceedings of the IEEE International Conference on Computer Vision*, 2011. 7

[22] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[23] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *Proceedings of the European Conference on Computer Vision*, 2014. 2

[24] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 2, 3, 5, 6

[25] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016. 2, 5

[26] 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. 8

[27] P. Krähenbühl and V. Koltun. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Proceedings of Neural Information Processing Systems*, 2012. 2, 7

[28] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. *arXiv preprint arXiv:1603.07063*, 2016. 6

[29] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. *arXiv preprint arXiv:1511.04510*, 2015. 6

[30] G. Lin, C. Shen, A. van den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6, 7, 8

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ra-

manan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014. 7

[32] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011. 8

[33] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2

[34] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 1, 2

[35] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 7

[36] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2, 3, 6, 7, 8

[37] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, et al. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 8

[38] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 1, 2, 7

[39] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 8

[40] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 2

[41] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision*, 2012. 7

[42] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015. 1

[43] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 8

[44] A. Vedaldi and K. Lenc. MatConvNet – convolutional neural networks for matlab, 2014. 6

[45] F. Xia, P. Wang, L.-C. Chen, and A. L. Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. *arXiv preprint arXiv:1511.06881*, 2015. 6

[46] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, 2015. 7, 8

[47] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2, 7

[48] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20K dataset. *CoRR*, abs/1608.05442, 2016. 8