On the iterative refinement of densely connected representation levels for semantic segmentation

Arantxa Casanova[1,2] Guillem Cucurull[1,2] Michal Drozdzal[1,3] Adriana Romero[1,3] Yoshua Bengio[1]

[1] Montreal Institute for Learning Algorithms   [2] Computer Vision Center, Barcelona

[3] Facebook AI Research

Abstract

State-of-the-art semantic segmentation approaches increase the receptive field of their models by using either a downsampling path composed of poolings/strided convolutions or successive dilated convolutions. However, it is not clear which operation leads to best results. In this paper, we systematically study the differences introduced by distinct receptive field enlargement methods and their impact on the performance of a novel architecture, called Fully Convolutional DenseResNet (FC-DRN). FC-DRN has a densely connected backbone composed of residual networks. Following standard image segmentation architectures, receptive field enlargement operations that change the representation level are interleaved among residual networks. This allows the model to exploit the benefits of both residual and dense connectivity patterns, namely: gradient flow, iterative refinement of representations, multi-scale feature combination and deep supervision. In order to highlight the potential of our model, we test it on the challenging CamVid urban scene understanding benchmark and make the following observations: 1) downsampling operations outperform dilations when the model is trained from scratch, 2) dilations are useful during the finetuning step of the model, 3) coarser representations require less refinement steps, and 4) ResNets (by model construction) are good regularizers, since they can reduce the model capacity when needed. Finally, we compare our architecture to alternative methods and report state-of-the-art result on the Camvid dataset, with at least twice fewer parameters.

1. Introduction

Convolutional Neural Networks (CNNs) have been extensively studied in the computer vision literature to tackle a variety of tasks, such as image classification [14, 17, 16], object detection [12] and semantic segmentation [21, 6]. Major advances have been driven by novel *very deep* architectural designs [14, 17], introducing skip connections to facilitate the forward propagation of relevant information

to the top of the network, and provide shortcuts for gradient flow. Very deep architectures such as residual networks (ResNets) [14], densely connected networks (DenseNets) [17] and squeeze-and-excitation networks [16] have exhibited outstanding performance on standard large scale computer vision benchmarks such as ImageNet [39] and MSCOCO [28].

Among top performing classification networks, ResNets challenge the hierarchical representation learning view of CNNs [26, 42, 11]. The hierarchical representation view associates the layers in network to different levels of abstraction. However, contrary to previous architectures such as [40], dropping or permuting almost any layer in a ResNet has shown to only minimally affect their overall performance [42], suggesting that the operations applied by a single layer are only a small modification to the identity operation. Significant effort has been devoted to analyzing and understanding these findings. On one hand, it has been argued that ResNets behave as an ensemble of shallow networks, averaging exponentially many subnetworks, which use different subsets of layers [42]. On the other hand, it has been suggested that ResNets engage in an unrolled iterative estimation of representations, that refine upon their input [11]. These arguments have been exploited in [8] to learn normalized inputs for iterative estimation, highlighting the importance of having transformations prior to the residual blocks.

Fully Convolutional Networks (FCNs) were presented in [30, 38] as an extension of CNNs to address per pixel prediction problems, by endowing standard CNNs with an upsampling path to recover the input spatial resolution at their output. In the recent years, FCN counterparts and enhanced versions of top performing classification networks have been successfully introduced in the semantic segmentation literature. Fully Convolutional ResNets (FC-ResNets) were presented and analyzed in [9] in the context of medical image segmentation. Moreover, Fully Convolutional DenseNets (FC-DenseNets) [21] were proposed to build low capacity networks for semantic segmentation, taking advantage of iterative concatenation of features maps.

In this paper, we further exploit the iterative refinement properties of ResNets to build densely connected residual networks for semantic segmentation, which we call Fully Convolutional DenseResNets (FC-DRNs). Contrary to FC-DenseNets [21], where the convolution layers are densely connected, FC-DRN apply dense connectivity to ResNets models. Thus, our model performs iterative refinement at each representation level (in a single ResNet) and uses dense connectivity to obtain refined multi-scale feature representations (from multiple ResNets) in the pre-softmax layer. We demonstrate the potential of our architecture on the challenging CamVid [4] urban scene understanding benchmark and report state-of-the-art results. To compare and contrast with common pipelines based on top performing classification CNNs, we perform an in depth analysis on different downsampling operations used in the context of semantic segmentation: dilated convolution, strided convolution and pooling. Although dilated convolutions have been well adopted in the semantic segmentation literature, we show that such operations seem to be beneficial only when used to finetune a pre-trained network that applies downsampling operations (e.g. pooling or strided convolution). When trained from scratch, dilation-based models are outperformed by their pooling and strided convolutions-based counterparts, highlighting the generalization capabilities of downsampling operations.

The contributions of our paper can be summarized as:

- We combine FC-DenseNets and FC-ResNets into a single model (FC-DRN) that fuses the benefits of both architectures: gradient flow and iterative refinement from FC-ResNets as well as multi-scale feature representation and deep supervision from FC-DenseNets.

- We show that FC-DRN model achieves state-of-the-art performance on CamVid dataset [4]. Moreover, FC-DRN outperform FC-DenseNets, while keeping the number of trainable parameters small.

- We provide an analysis on different operations enlarging the receptive field of a network, namely poolings, strided and dilated convolutions. We inspect FC-DRN by dropping ResNets from trained models as well as by visualizing the norms of the weights of different layers. Our experiments suggest that *the benefits of dilated convolutions only apply when combined with pre-trained networks that contain downsampling operations*. Moreover, we show that *ResNets (by model construction) are good regularizers, since they can reduce the model capacity at different representation levels when needed, and adapt the refinement steps*.

2. Related work

In the recent years, FCNs have become the *de facto* standard for semantic segmentation. Top performing classifica-

tion networks have been successfully extended to perform semantic segmentation [43, 34, 9, 44, 21].

In order to overcome the spatial resolution loss induced by successive downsampling operations of classification networks, several alternatives have been introduced in the literature; the most popular ones being long skip connections in encoder-decoder architectures [30, 2, 38, 19] and dilated convolutions [46, 7]. Long skip connections help recover the spatial information by merging features skipped from different resolutions on the contracting path, whereas dilated convolutions enlarge the receptive field without downsizing the feature maps.

Another line of research seeks to endow segmentation pipelines with the ability to enforce structure consistency to their outputs. The contributions in this direction include Conditional Random Fields (CRFs) and its variants (which remain a popular choice) [24, 7, 49], CRFs as Recurrent Neural Networks [49], iterative inference denoising autoencoders [37], convolutional pseudo-priors [45], as well as graph-cuts, watersheds and spatio-temporal regularization [3, 34, 25].

Alternative solutions to improve the performance of segmentation models are based on combining features at different levels of abstraction. Efforts in this direction include iterative concatenation of feature maps [17, 21]; fusing up-sampled feature maps with different receptive fields prior to the softmax classifier [5], along the network [27, 1] or by means of two interacting processing streams operating at different resolutions [33]; gating skip connections between encoder and decoder to control the information to recover [19]; and using a pyramid pooling module with different spatial resolutions for context aggregation [48]. Moreover, incorporating global features has long shown to improve semantic segmentation performance [10, 29].

Finally, semantic segmentation performance has also been improved by training with synthetic data [35], propagating information through video data [20], or modeling uncertainties in the model [23].

3. Fully Convolutional DenseResNet

In this section, we briefly review both ResNets and DenseNets, and introduce the FC-DRN architecture.

3.1. Background

Let us denote the feature map representation of the l-th layer of the model as x_l. Traditionally, in CNNs, the feature map x_l is obtained by applying a transformation H, composed of a convolution followed by a non-linearity, to the $l - 1$-th feature map x_{l-1} as $x_l = H(x_{l-1})$. CNNs are built by stacking together multiple such transformations. However, due to the non-linearity operation, optimization of such networks becomes harder with growing depth. Architectural solutions to this problem have been proposed in
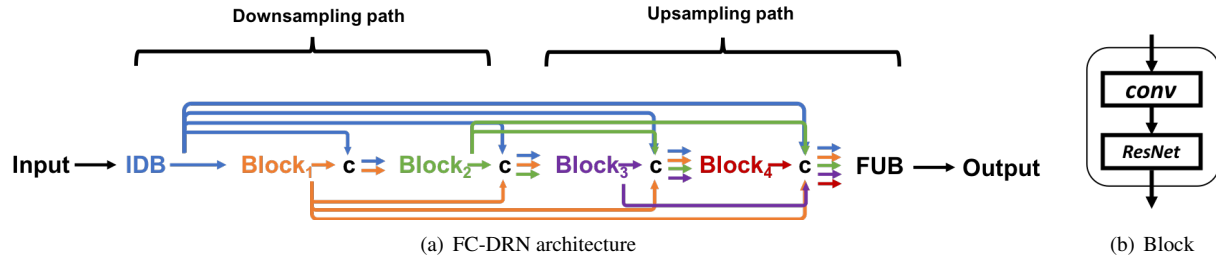
(a) FC-DRN architecture

(b) Block

Figure 1: (a) Diagram representing our model, a dense connectivity with four Residual Networks. We use color coding to indicate the connectivity pattern in our model. IDB stands for Initial Downsampling Block and FUB stands for Final Upsampling Block. The outputs of consecutive operations are concatenated (letter c on the figure). Whenever needed, we apply transformations over the paths marked with arrows. In the downsampling path, we apply either pooling operation, strided convolution or dilated convolution to increase the network's receptive field. In the upsampling path, we apply upsampling operations to compensate for pooling or strided convolutions and 1×1 convolution for dilations. (b) Each *Block* contains a convolution operation followed by a Residual Network.

ResNets [14] and DenseNets [17].

In ResNets, the representation of l-th feature map is obtained by learning the residual transformation H of the input feature map x_{l-1} and summing it with the input x_{l-1}. Thus, the l-th feature map representation can be computed as follows: $x_l = H(x_{l-1}) + x_{l-1}$. This simple modification in network's connectivity introduces a path that has no non-linearities, allowing to successfully train networks that have hundreds (or thousands) of layers. Moreover, lesion studies performed on ResNets have opened the door to research directions that try to better understand how these networks work. Following these lines, it has been suggested that ResNets layers learn small modifications of their input (close to the identity operation), engaging in an iterative refinement of their input.

In DenseNets, the l-th feature map is obtained by applying a transformation H to all previously obtained feature maps such that $x_l = H([x_0, x_1, \cdots, x_{l-1}])$, where $[\cdot, \cdot]$ denotes the concatenation operation. One can easily notice that when following the dense connectivity pattern in DenseNets, the pre-softmax layer receives the concatenation of all previous feature maps. Thus, DenseNets introduce deep feature supervision by means of their model construction. It has been shown that using the deep connectivity pattern one can train very deep models that outperform ResNets [17]. Moreover, it is worth mentioning that combining information at different representation levels has shown to be beneficial in the context of semantic segmentation [10, 21].

3.2. FC-DRN model

FC-DRNs extend the FC-DenseNets architecture of [21] and incorporate a dense connectivity pattern over multiple

ResNets[1]. Thus, FC-DRNs combine the benefits of both architectures: FC-DRNs perform iterative estimation at each abstraction level (by using ResNets) and combine different abstraction levels while obtaining deep supervision (by means of DenseNets connections).

The connectivity pattern of FC-DRN is visualized in Figure 1. First, the input is processed with an Initial Downsampling Block (IDB) composed of a single convolution followed by 2×2 pooling operation and two 3×3 convolutions. Then, the output is fed to a *dense block* (the densely connected part of the model), which is composed of ResNets, transformations and concatenations, forming a downsampling path followed by an upsampling path.

In our model, there are 9 ResNets, motivated by the standard number of downsampling and upsampling operations in the FCN literature. Each ResNet is composed of 7 basic blocks, computing twice the following operations: batch normalization, ReLU activation, dropout and 3x3 convolution. After each ResNet, we apply a *transformation* with the goal of *changing the representation level*. This transformation is different in the downsampling and upsampling paths: in the downsampling path, it can be either a pooling, a strided convolution or a dilated convolution; whereas in the upsampling path, it can be either an upsampling to compensate for pooling/strided convolution or a 1×1 convolution in case of dilated convolutions, to keep models with roughly the same capacity. Following [7, 6], transformations in the dilation-based model adopt a multi-grid pattern.

The outputs of the transformations are concatenated such that the input to the subsequent ResNet incorporates information from all the previous ResNets. Concatenations are performed over channel dimensions and, if needed, the resolution of the feature maps is adjusted using transforma-

---

[1]Note that in [21] the dense connectivity pattern is over convolutional operations.

tions that are applied independently to each concatenation input[2]. After each concatenation, there is a 1×1 convolution to mix the features. Finally, the output of the dense block is fed to a Final Upsampling Block (FUB) that adapts the spatial resolution and the number of channels in the model output.

4. Analysis and Results

In this section, we assess the influence of applying different kinds of transformations between different ResNets and report our final results.

All experiments are conducted on the CamVid [4] dataset, which contains images of urban scene. Each image has a resolution of 360×480 pixels and is densely labeled with 11 semantic classes. The dataset consists of 367, 101 and 233 frames for training, validation and test, respectively. In order to compare different architectures, we report results on the validation set with two metrics: mean intersection over union (mean IoU) and global accuracy.

All networks were trained following the same procedure. The weights were initialized with HeUniform [13], and the networks were trained with RMSProp optimizer [41], with a learning rate of $1e - 3$ and an exponential decay of 0.995 after each epoch. We used a weight decay of $1e - 4$ and dropout rate of 0.2. The dataset was augmented with horizontal flipping and crops of 324x324. We used early stopping on the validation mean IoU metric to stop the training, with a patience of 200 epochs.

4.1. FC-DRN transformation variants

State-of-the-art classification networks downsample their feature maps' resolution by successively applying pooling (or strided convolution) operations. In order to mitigate the spatial resolution loss induced by such subsampling layers, many segmentation models only allow for a number of subsampling operations and change the remaining ones by dilated convolutions [7, 46, 47]. However, in some other cases [21, 31, 38], the number of downsampling operations is preserved, recovering fine grained information from via long skip connections. Therefore, we aim to analyze and compare the influence of pooling/upsampling operations versus dilated convolutions. To that aim, we build sister architectures, which have an initial downsampling block, followed by a dense block, and a final upsamping block, as described in Section 3.2, and only differ in the transformation operations applied within their respective dense blocks.

Max-Pooling architecture (FC-DRN-P): This architecture interleaves ResNets with four max-pooling operations (downsampling path) and four nearest neighbor up-

samplings followed by 3x3 convolutions to smooth the output (upsampling path).

Strided convolution architecture (FC-DRN-S): This architecture interleaves ResNets with four strided convolution operations (downsampling path) and four nearest neighbor upsamplings followed by 3x3 convolutions to smooth the output (upsampling path).

Dilated architecture (FC-DRN-D): This architecture interleaves ResNets with four multi-grid dilated convolution operations of increasing dilation factor (2, 4, 16 and 32)[3]. and standard convolutions to emulate the upsampling operations. Note that the dense block of this architecture does not change the resolution of its feature maps.

FC-DRN-P finetuned with dilations (FC-DRN-P-D): This architecture seeks to mimic state-of-the-art models based on top performing classification networks, which replace the final subsampling operations with dilated convolutions [7, 46, 47]. More precisely, we substitute the last two pooling operations of FC-DRN-P by dilated convolutions of dilation rate 4 and 8, respectively. Following the spirit of FC-DRN-D, the first two upsampling operations become standard convolutions. We initialize our dilated convolutions to the identity, as suggested in [46].

FC-DRN-S finetuned with dilations (FC-DRN-S-D): Following FC-DRN-P-D, we substitute the last two strided convolution operations of FC-DRN-S by dilated convolutions (rates 4 & 8), whereas the first two upsampling operations become standard convolutions. In this case, we initialize the weights of the dilated convolutions with the weights of the corresponding pre-trained strided convolutions.

Table 1 reports the validation results for the described architectures. Among the networks trained from scratch, FC-DRN-P achieves the best performance in terms of mean IoU by a margin of 0.8% and 3.7% w.r.t. FC-DRN-S and FC-DRN-D, respectively. When finetuning the pooling and strided convolution architectures with dilations, we further improve the results to 81.7% and 81.1%, respectively. It is worth noting that we also tried training FC-DRN-P-D and FC-DRN-S-D from scratch, which yielded worse results, highlighting the benefits of pre-training with poolings/strided convolutions, which capture larger contiguous contextual information.

Figure 2 presents qualitative results from all architectures. As illustrated in the figure, FC-DRN-P prediction seems to better capture the global information, when compared to FC-DRN-D. This can be observed on the left part of the predictions, where dilated convolutions predict different classes for isolated pixels. Max-poolings understand better the scene and output a cleaner and more consistent prediction. Although FC-DRN-P has a more global view of the scene and is less prone to make local mistakes, it lacks res-

---

[2]Note that in order to maintain the number of transformations when comparing different models (e.g. pooling-based vs. dilation-based model), we apply a convolution even when concatenating same resolution feature maps.

[3]We tested many different variants of dilation factors and found out that this multi-grid structure gives the best results.

Architecture	mean IoU [%]	accuracy [%]
FC-DRN-P	**81.1**	**96.1**
FC-DRN-S	80.3	95.9
FC-DRN-D	77.4	95.5
FC-DRN-P-D	**81.7**	**96.0**
FC-DRN-S-D	81.1	96.0

Table 1: Comparison of FC-DRN architectures with different transformations. Results reported on the validation set.

olution to make fine-grained predictions. In the FC-DRN-P-D prediction, we can see that dilated convolutions help recover the column poles in the middle of the image that were not properly identified in the other architectures, and the predictions of the pedestrians on the left are also sharper. However, the model still preserves the global information of FC-DRN-P, reducing some errors that were present on the left part of the image in the FC-DRN-D prediction. If we take a look at the FC-DRN-S prediction, we can see that it recovers the pedestrian on the right successfully, but fails to correctly segment the sidewalk and the pedestrians on the left. Finetuning this architecture with dilations (FC-DRN-S-D) helps capture a lost pedestrian on the left, but still lacks the ability to sharply sketch the right column pole. Furthermore, there are some artifacts on the top part of the image in both strided convolution-based architectures.

4.2. Results

Following the comparison in Table 1, we report our final results on the FC-DRN-P-D architecture. Recall that this architecture is a pre-trained FC-DRN with a dense block of 4 max pooling operations (downsampling path) and 4 repeat and convolve operations (upsampling path) and finetuned by substituting the last two max poolings and the first two upsamplings by dilated convolutions, on the same data.

Table 2 compares the performance of our model to state-of-the-art models. As shown in the table, our FC-DRN-P-D exhibits state-of-the-art performance when compared to previous methods, especially when it comes to segmenting under-represented classes such as column poles, pedestrians and cyclists. It is worth noting that our architecture improves upon pre-trained models with 10 times more parameters. When compared to FC-DenseNets, FC-DRN outperform both FC-DenseNet67 (with a comparable number of parameters) and FC-DenseNet103 (with only 41.5% of its parameters) by 2.5% and 1.4% mean IoU, respectively. Moreover, it also exceeds the performance of more recent methods such as G-FRNet, which uses gated skip connections between encoder and decoder in a FCN architecture, while only using 13% of its parameters.

In order to further boost the performance of our network, we finetuned it by using soft targets [15, 36] (0.9 and 0.01, instead of 1 and 0 in the target representation), improving generalization and obtaining a final score of 69.4% mean IoU and 91.6% global accuracy. Using soft targets allows

the network to become more accurate in predicting classes such as pedestrian, fence, column pole, sign, building and sidewalk when compared to the original version. FC-DRN recovers slim objects such as column poles and pedestrians much better than other architectures presented in the literature, while maintaining a good performance on classes composed of larger objects.

It is worth mentioning that, unlike most of current state-of-the-art methods, FC-DRN have not been pre-trained on large datasets such as ImageNet [39]. Moreover, there are other methods in the literature that exploit virtual images to augment the training data [35] or that leverage temporal information to improve performance [20]. Note that those enhancements complement each other and FC-DRN could most likely benefit from them to boost their final performance as well. However, we leave those as future work.

Figure 3 shows some FC-DRN segmentation maps (right) compared to their respective ground truths (middle). We can observe that the segmentations have good quality, aligned with the quantitative results we obtained. The column poles and pedestrians are sharply segmented, but some difficulties arise when trying to distinguish between sidewalks and roads or in the presence of small road signs.

5. Delving deeper into FC-DRN transformations

In this section, we provide an in depth analysis of the variants of the trained FC-DRN architectures to compare different transformation operations: pooling, dilation and strided convolution. We start by dropping ResNets from a FC-DRN and then look into the weight's norms of all ResNets in the models. We end the section with a discussion exploiting the observations from the network inspection.

We follow the strategy of dropping layers introduced in [42, 18], and drop all residual blocks of a ResNet (we only keep the first residual block that adjusts the depth of the feature maps) with the goal of analyzing the implications of using different transformation operations. The results of the experiment are shown in Figure 4. On one hand, Figure 4(a) reports the performance drops in percentage of mean IoU for each ResNet in the initial networks (i.e. FC-DRN-P, FC-DRN-D and FC-DRN-S). Surprisingly, dropping ResNets 3 to 8 barely affects the performance of FC-DRN-D. However, both pooling and strided convolution models suffer from the loss of almost any ResNet. On the other hand, Figure 4(b) presents the results of dropping ResNets in the finetuned models (i.e. FC-DRN-P-D and FC-DRN-P-S). Finetuning the pooling network with dilations makes ResNets 5 and 6 slightly less necessary, while ResNet 8 becomes the most relevant one. Finetuning the strided convolution network with dilations makes ResNet 1 extremely important, while ResNets 4 to 6 have a smaller influence. In general, it seems that finetuning with dilations reduces the importance
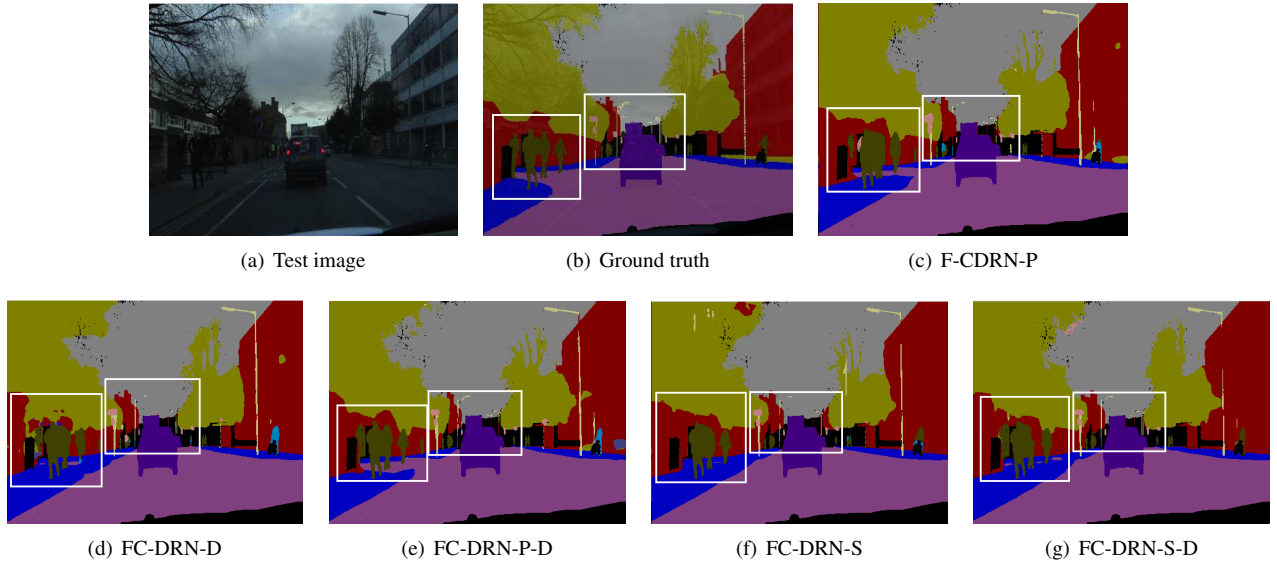
Figure 2: Qualitative results on the test set: (a) input image, (b) ground truth, (c) FC-DRN-P prediction, (d) FC-DRN-D prediction, (e) FC-DRN-P-D prediction, (f) FC-DRN-S prediction and (g) FC-DRN-S-D prediction. Main differences are highlighted with white boxes.

Model	# params [M]	Ext. Data	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Column pole	Sidewalk	Cyclist	mean IoU [%]	Gl. acc. [%]
SegNet [2]	29.5	Yes	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	46.4	62.5
DeconvNet [31]	252	Yes						n/a						48.9	85.9
FCN8 [30]	134.5	Yes	77.8	71.0	88.7	76.1	32.7	91.2	41.7	24.4	19.9	72.7	31.0	57.0	88.0
Visin et al. [43]	32.3	Yes						n/a						58.8	88.7
DeepLab-LFOV [7]	37.3	Yes	81.5	74.6	89.0	82.2	42.3	92.2	48.4	27.2	14.3	75.4	50.1	61.6	-
Bayesian SegNet [22]	29.5	Yes						n/a						63.1	86.9
Dilation8 [46]	140.8	Yes	82.6	76.2	89.0	84.0	46.9	92.2	56.3	35.8	23.4	75.3	55.5	65.3	79.0
FC-DenseNet67 [21]	3.5	No	80.2	75.4	93.0	78.2	40.9	**94.7**	58.4	30.7	38.4	81.9	52.1	65.8	90.8
Dilation8 + FSO [25]	130	Yes	**84.0**	77.2	91.3	**85.6**	**49.9**	92.5	59.1	37.6	16.9	76.0	57.2	66.1	88.3
FC-DenseNet103 [21]	9.4	No	83.0	**77.3**	**93.0**	77.3	43.9	94.5	59.6	37.1	37.8	82.2	50.5	66.9	91.5
G-FRNet [19]	30	Yes	82.5	76.8	92.1	81.8	43.0	94.5	54.6	**47.1**	33.4	**82.3**	59.4	68.0	90.8
FC-DRN -P-D	3.9	No	82.6	75.7	92.6	79.9	42.3	94.1	<u>61.2</u>	36.9	<u>42.6</u>	81.2	**61.8**	<u>68.3</u>	91.4
FC-DRN-P-D + ST	3.9	No	83.5	75.6	92.1	78.5	46.6	93.9	**62.7**	44.3	**43.1**	82.2	60.8	**69.4**	**91.6**

Table 2: Results on CamVid test set, reported as IoU per class, mean IoU and global accuracy, compared to state-of-the-art.
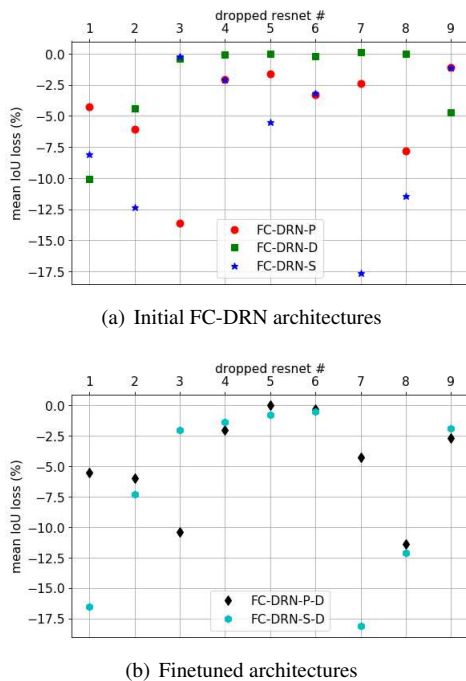
of the bottleneck ResNets of the network. Moreover, the results might suggest that dilations do not change the representation level as much as poolings/strided convolutions.

To gain further insight on what FC-DRN variants are learning, we visualize the ℓ_1 norm of the weights in all ResNets. More precisely, given a $4D$ weight tensor w of a shape $[N, M, K, K]$ that applies the transformation between two consecutive layers with N and M channels, respectively, we compute: $1/M \sum_m ||w[:, m, :, :]||_1$. The results of this experiment for different FC-DRN variants are shown in Figure 5. The weight norms shown in the figure are in line with the discussion of Figure 4. On one hand, basic FC-DRN architectures are displayed in Figure 5(a). For FC-DRN-P, we can see how the weights in ResNet 5

have lower values, suggesting an explanation for the lower drop in performance when removing ResNet 5. Furthermore, we can also see that the weight norms in the case of FC-DRN-D are almost zero for ResNets 3 to 8, suggesting that the network does not fully exploit its capacity. In the case of FC-DRN-S, the norms of the weights in ResNet 5 are smaller (similar to FC-DRN-P), whereas ResNet 2 exhibits some of the highest norms. On the other hand, Figure 5(b) shows the results of finetuning. FC-DRN-P-D has weight norms among ResNets 4 to 6 lower than FC-DRN-P, whereas FC-DRN-S-D follows a similar pattern when compared to FC-DRN-S. However, from the observed weights, it would seem as ResNet 4 still benefits from some refinement steps. Overall, it seems that finetuning consistently

Figure 3: Qualitative results on the CamVid test set. Left: images, middle: ground truths, right: FC-DRN predictions.

(a) Initial FC-DRN architectures
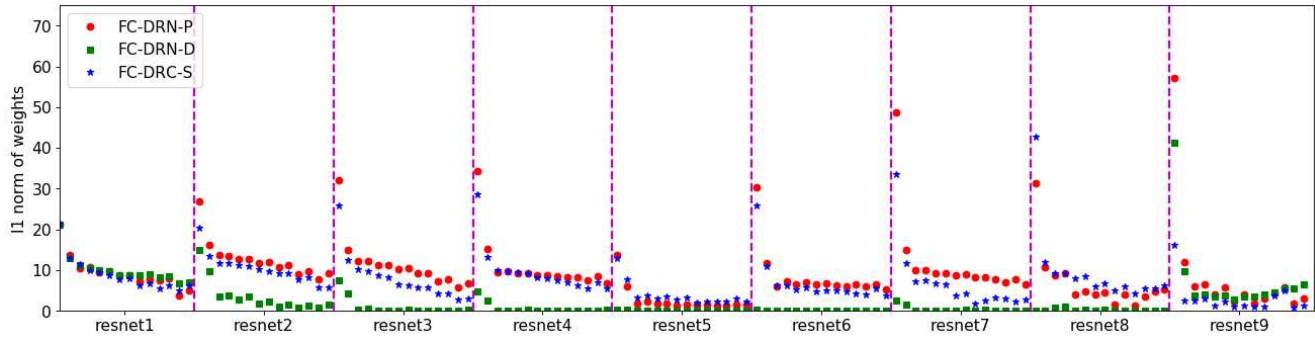

(b) Finetuned architectures

Figure 4: Results of dropping ResNets from a trained FC-DRN reported for the validation set: y-axis represents the loss in mean IoU when comparing to the model with all ResNets, x-axis represents the ID of the dropped ResNet.
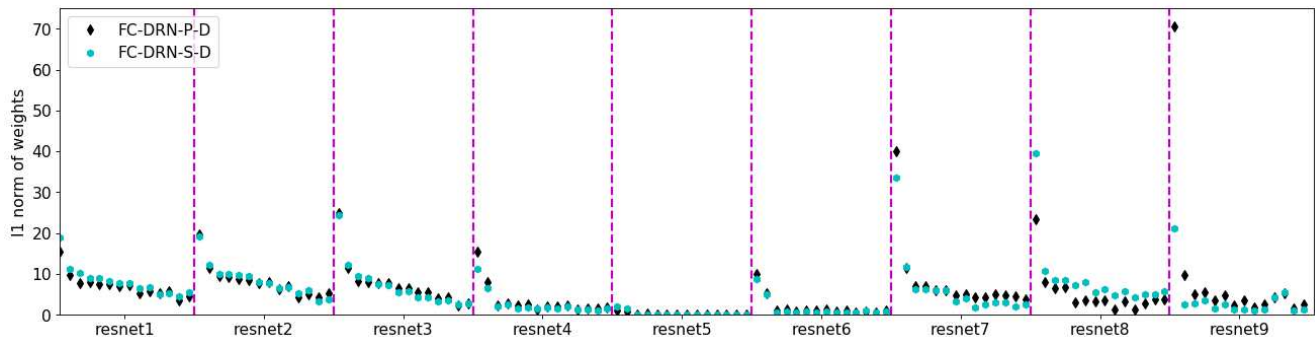
reduces the weight norms of the bottleneck ResNets.

It is important to note that the structure of ResNets, due to the usage of the residual block, allows the model to self-adjust its capacity when needed, forcing the residual transformation of the residual block to be close to 0 and using the identity connection to forward the information. We hypothesize that this behavior of residuals is observed for some layers in our model (as it is shown in Figure 5). To test our

hypothesis, we decided to reduce the capacity of *trained* FC-DRN by removing layers from the residuals of ResNets for which the norm of weights is small and to monitor the performance of the *compressed* FC-DRN model. If our hypothesis is true, then removing the residuals in the layers where the norm is close to 0 should not affect strongly the model performance. We choose to drop the layers whose weight norms are close enough to zero, based on visual inspection of Figure 5, thus allowing each representation level to have different number of refinement steps. The results of this trained model compression experiment are reported in Table 3. We can see that after removing 8% of the parameters from FC-DRN-P and FC-DRN-S, there is a drop in mean IoU on validation set of -1.6 and -5.4, respectively. Interestingly, we were able to remove 38% of weights from FC-DRN-D model while only experiencing a drop of -0.8 in mean IoU. Both finetuned models can be compressed by removing 15% of the capacity with slight performance drops of -1 and -1.7 for FC-DRN-P-D and FC-DRN-S-D, respectively. In general, it seems that finetuning the models with dilations not only improves the segmentation results but also makes the model more compressible.

Finally, we test if the optimization process of the high capacity FC-DRN reaches better local minima, due to self-adjustment of ResNets' capacity, than if we train a low capacity FC-DRN from scratch. To this end, we trained from scratch the reduced capacity FC-DRN-D model and compared the results to the numbers reported in Table 3. The re-trained model obtained the mean validation IoU of 76.6. This is 0.8% below the result reported for the high capacity FC-DRN-D. We hypothesize that the model capacity reduction during the optimization process helps in reaching a better local minima. Since the self-adjustment of the ResNet capacity might be encouraged by weight decay, for the sake of the comparison we also trained the reduced ca-

(a) Initial FC-DRN architectures

(b) Finetuned architectures

Figure 5: Norms of the weights in ResNets from trained FC-DRN variants. Different ResNets separated with vertical lines.

Architecture	mean IoU loss [%]	compression rate
FC-DRN-P	−1.6	1.08
FC-DRN-D	−0.8	1.38
FC-DRN-S	−5.4	1.08
FC-DRN-P-D	−1	1.15
FC-DRN-S-D	−1.7	1.15

Table 3: Performance drop for different models when removing all layers with small norm of weights from a trained FC-DRN variant. Compression rate represents the ratio between the number of parameters in the original model and the number of parameters after removing layers. Results reported on the validation set.

pacity model without weight decay at all. This model obtained the mean validation IoU of 75.4% that is 2% below the full capacity model.

6. Conclusions

In this paper, we combined two standard image segmentation architectures (FC-DenseNets and FC-ResNets) into a single network that we called Fully Convolutional DenseResNet. Our FC-DRN fuses the benefits of both models: gradient flow and iterative refinement from FC-ResNets as well as multi-scale feature representation and deep supervision from FC-DenseNets. We demonstrated the potential of our model on the challenging CamVid urban scene understanding benchmark and reported state-of-the-art results, with at least 2x fewer parameters than concurrent models.

Additionally, we analyzed different downsampling operations used in the context of semantic segmentation: dilated convolution, strided convolution and pooling. We inspected the FC-DRN by dropping ResNets from the trained models as well as by visualizing the weight norms of different layers and showed that ResNets (by model construction) are good regularizers, since they can reduce the model capacity when needed. In this direction, we observed that coarser representations seem to benefit from less refinement steps. Moreover, our results comparing different transformations suggest that pooling offers the best generalization capabilities, while the benefits of dilated convolutions only apply when combined with pre-trained networks that contain downsampling operations.

References

[1] I. Ardiyanto and T. B. Adji. Deep residual coalesced convolutional network for efficient semantic road segmentation. *MVA*, 2017.

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

[3] T. Beier, B. Andres, U. Köthe, and F. A. Hamprecht. An efficient fusion move algorithm for the minimum cost lifted multicut problem. In *Lecture Notes in Computer Science*. 2016.

[4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, 2008.

[5] H. Chen, X. Qi, J.-z. Cheng, and P.-a. Heng. Deep contextual networks for neuronal structure segmentation. *AAAI*, 2016.

[6] L. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, 2017.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv:1606.00915*, 2016.

[8] M. Drozdzal, G. Chartrand, E. Vorontsov, M. Shakeri, L. Di Jorio, A. Tang, A. Romero, Y. Bengio, C. Pal, and S. Kadoury. Learning normalized inputs for iterative estimation in medical image segmentation. *Medical image analysis*, 2018.

[9] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal. The importance of skip connections in biomedical image segmentation. *Deep Learning and Data Labeling for Medical Applications*, 2016.

[10] C. Gatta, A. Romero, and J. van de Veijer. Unrolling Loopy Top-Down Semantic Feedback in Convolutional Deep Networks. In *CVPRW*, 2014.

[11] K. Greff, R. K. Srivastava, and J. Schmidhuber. Highway and residual networks learn unrolled iterative estimation. *ICLR*, 2017.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *arXiv preprint arXiv:1703.06870*, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *ICCV*, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *CVPR*, 2016.

[15] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. *Deep Learning workshop at NIPS*, 2014.

[16] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[17] G. Huang, Z. Liu, and K. Q. Weinberger. Densely Connected Convolutional Networks. *CVPR*, 2017.

[18] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. *CoRR*, 2016.

[19] M. A. Islam, M. Rochan, N. D. Bruce, and Y. Wang. Gated feedback refinement network for dense image labeling. In *CVPR*, 2017.

[20] V. Jampani, R. Gadde, and P. V. Gehler. Video Propagation Networks. *CVPR*, 2017.

[21] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *CVVT, CVPRW*, 2017.

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

[23] A. Kendall and Y. Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *NIPS*, 2017.

[24] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*. 2011.

[25] A. Kundu, V. Vineet, and V. Koltun. Feature space optimization for semantic video segmentation. In *CVPR*, 2016.

[26] Q. Liao and T. A. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *CoRR*, 2016.

[27] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. *CVPR*, 2017.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[29] W. Liu, A. Rabinovich, and A. C. Berg. ParseNet: Looking Wider to See Better. *arXiv:1506.04579*, jun 2015.

[30] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *CVPR*, 2015.

[31] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. *arXiv preprint arXiv:1505.04366*, 2015.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[33] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. *CVPR*, 2017.

[34] T. M. Quan, D. G. Hilderbrand, and W.-K. Jeong. Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics. *arXiv preprint arXiv:1612.05360*, 2016.

[35] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. *ECCV*, 2016.

[36] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

[37] A. Romero, M. Drozdzal, A. Erraqabi, S. Jégou, and Y. Bengio. Image segmentation by iterative inference from conditional score estimation. *arXiv preprint arXiv:1705.07450*, 2017.

[38] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, 2015.

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[40] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR*, 2015.

[41] T. Tieleman and G. Hinton. rmsprop adaptive learning. In *COURSERA: Neural Networks for Machine Learning*, 2012.

[42] A. Veit, M. Wilber, and S. Belongie. Residual Networks Behave Like Ensembles of Relatively Shallow Networks. *NIPS*, 2016.

[43] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville. ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation. *CVPR workshop*, 2016.

[44] Z. Wu, C. Shen, and A. van den Hengel. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *arXiv:1611.10080*, nov 2016.

[45] S. Xie, X. Huang, and Z. Tu. *Top-Down Learning for Structured Labeling with Convolutional Pseudoprior*, pages 302–317. Springer International Publishing, Cham, 2016.

[46] F. Yu and V. Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. *ICLR*, 2016.

[47] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. *CVPR*, 2017.

[48] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *CVPR*, 2017.

[49] 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. *ICCV*, 2015.