DPDB-Net: Exploiting Dense Connections for Convolutional Encoders

Gabriel L. Oliveira, Wolfram Burgard and Thomas Brox

***Abstract*— Densely connected networks for classification enable feature exploration and result in state-of-the-art performance on multiple classification tasks. The alternative to dense networks is the residual network which enables feature re-usage. In this work, we combine these orthogonal concepts for encoder-decoder architectures, which we call Dual-Path Dense-Block Network (DPDB-Net). We introduce a dense block which incorporates feature re-usage and new feature exploration in the encoder. Moreover, we discuss that feature re-usage by the residual network architecture leads to a feature map explosion in the decoder and, thus, is not advantageous in this part of the network. We evaluated our proposed architecture in multiple segmentation tasks and report state-of-the-art performance on the Freiburg Forest dataset and competitive results on the CamVid dataset.**

I. Introduction

Convolutional Neural Networks (CNNs) are in the center of major advances in several areas of computer vision and robotics, such as image classification [12], [13], [16], [28], loop closure [11], optical flow [9], localization [29] and image segmentation [21]. The advancement in segmentation tasks has been mainly due to encoder-decoder networks, also known under the term Fully Convolutional Networks (FCNs) [21], [27]. This kind of architecture extends classification CNNs, which only have an encoder, by adding a decoder layer which tackles pixel-wise prediction. The encoder part is typically a state-of-the-art classification architecture, such as VGG [28] or Resnet [12]. The encoder is responsible for feature learning and provides an initial low-resolution dense prediction. This low-resolution prediction is refined by the decoder, which consists of a set of convolutional layers with successive upsampling.

In order to recover the resolution loss induced by pooling layers, all current techniques make use of skip connections between their encoder and decoder parts. Skip connections between encoder and decoder help the upsampling path to recover fine-grained information from downsampling layers. Although with a different motivation, modern classification architectures like ResNet and DenseNet [12], [13] take advantage of skip connections, too, by propagating information from lower directly to higher layers within the encoder. Such connections facilitate gradient back-propagation to the bottom layers without magnitude reduction, thereby reducing the vanishing gradient problem.

ResNets introduced the residual function, which relies on a residual connection and is called residual path. Such connection performs element-wise summation between the input and output features of each micro-block. The residual unit has as main characteristic the reuse of features, and serves as a feature refinement stage. The recently introduced DenseNet [13] uses a new micro-block called *densely connected block*. In contrast to residual blocks which element-wise sum the input and output features through the residual path, the input features are concatenated with the output features permitting features in each micro-block to be connected to all previous blocks. The main effect of a *dense block* is that it keeps exploring new features. Additionally, DenseNets provide a higher parameter efficiency when compared to ResNets, which usually requires on average three times more

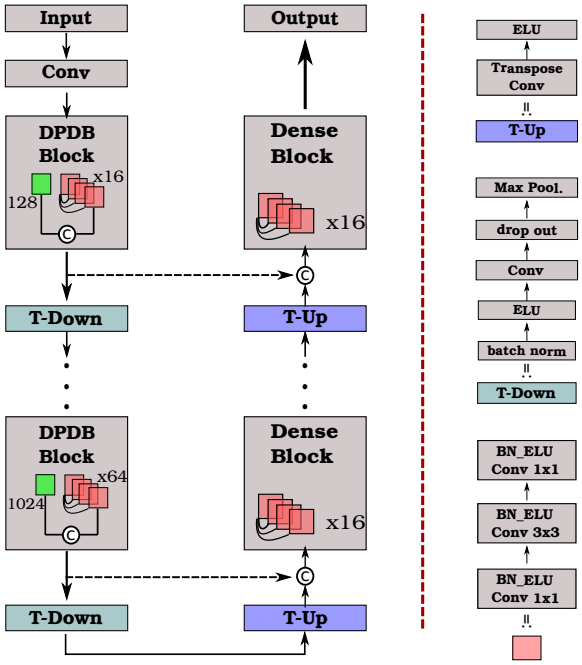

Fig. 1: Simplified diagram of the DPDB-Net. Our architecture is built from DPDB blocks, more specifically in the *encoder* section of the network. The *decoder* part uses dense blocks. Further discussion about this architectural decision is presented in Section III. The complete architecture consists of five DPDB blocks for encoding, and each of them is followed by a down sampling block, called *Transition-Down* (T-Down). The decoder part comprises five upsampling blocks, called *Transition-Up* (T-Up) blocks. The upsampled features are fused with its correspondent *encoder* part through concatenation and subsequently used as an input for a dense block. Skip connections are depicted as dotted horizontal lines and ⓒ represents a concatenation operation.

All authors are with the Department of Computer Science at the University of Freiburg, 79110 Freiburg, Germany. This work has been supported by the German Research Foundation (DFG) within the Cluster of Excellence BrainLinks-BrainTools (EXC 1086).

parameters to provide the same level of accuracy [13]. Such characteristic mitigates the feature map explosion problem, i.e., a computationally intractable number of feature maps with high resolution prior to the softmax layer.

In this work, we aim to incorporate feature refinement and exploration into a new block, called Dual-Path Dense-Block (DPDB). By fusing both characteristics, we are able to produce more informative features, for semantic segmentation tasks. Such fusing strategy has been explored before in [33], yet focusing on classification. We also propose a new network called DPDB-Net which provides a better understanding on dense connections for semantic segmentation tasks. An overview of the proposed method is shown in Fig. 1. The experimental evaluation shows that the proposed approach yields competitive results on the CamVid dataset [4] and state-of-the art results for the Freiburg Forest semantic segmentation dataset [31].

II. Related Works

In this section we review advances in semantic segmentation. Driven by the recent progress in classification with deep neural networks, pixel-level prediction achieved great success inspired by the fully convolutional concept of replacing fully-connected layers with convolutions [21]. Following the FCN concept many works tackle semantic segmentation through exploring context, resolution or boundary alignment. Adding context to FCNs is an active research area in semantic segmentation. Methods like Zoom-out [22], Parsenet [20] and Deeplab-V2 [6] were design to incorporate context explicitly. Zoom-out proposes a hierarchical context features network, while ParseNet includes global pooling features to explicitly add context information. More recently, Deeplab-V2 proposes Atrous Spatial Pyramid Pooling, which combines features at different fields of view given a set of dilated convolutions, to include context to a Resnet based encoder.

Other works on semantic segmentation focus on recovering the resolution lost by successive down-sampling operations, such as pooling. For instance Deconv-Net [23], Fast-Net [24] and SegNet [2] are examples of these approaches. Deconv-Net introduces an unpooling operation and an hourglass like network to learn the upsampling process, while SegNet reuses the pooling indices from the encoder to recover resolution. Fast-Net adds skip connections from the encoder features to the corresponding decoder activations also focusing on efficiency in terms of computational requirements.

Boundary approaches try to refine the predictions near the object edges. These approaches make use of post-processing techniques, such as Deeplab [5], Adelaide [18] and bilateral solver [3]. Deeplab make use of a CRF built on fully-connected graph, which serves as a boundary refinement after the CNN. Another example of CRFs as post-processing step is Adelaide [18]. Alternative solutions to CRFs are proposed by [3], [14]. [14] proposes the bilateral filter to learn specific potentials within CNNs, providing $10\times$ speed up and comparable performance to CRFs.

In contrary to previous works we will explore the potential of densely connected blocks, through our DPDB block, as an encoder-decoder architecture for semantic segmentation. Differently from previous approaches that use the same class of dense blocks homogeneously in the whole network, we found that different characteristics are required for the encoder and decoder part of FCNs. Our DPDB-Net is a new architecture that incorporates such demands for semantic segmentation tasks.

III. Methodology

We start with our observations on ResNets and DenseNets. They motivate the proposal of our DPDB block. Then using our block we design a fully-convolutional framework for semantic segmentation tasks. The goal of our model is to further exploit feature reuse and exploration.

A. Analysis of ResNet and DenseNet

In the following we will discuss and present an analysis on the strengths and weaknesses of the Resnet and DenseNet topologies. The findings provide the motivation behind developing an architecture that exploits of both approaches.

Let x_l be the output of the *l-th* layer. Standard CNNs compute x_l by applying a non-linear transformation N_l to the output of the previous layer x_{l-1}. The equation $x_l = N_l(x_{l-1})$ defines N_l as a set of operations, such as convolution followed by Exponential Linear Units (ELUs) [7] and dropout. Residual networks introduced the so-called *residual block* in order to ease the training of very deep architectures. The residual block sums the input and output layers:

$$x_l := N_l(x_{l-1}) + x_{l-1}, \tag{1}$$

making feature reuse possible and permitting gradients to flow directly to early layers. By sharing features across all steps, *residual blocks* encourage feature re-usage and thus reduce feature redundancy. This makes it more difficult for residual networks to explore new features. For residual blocks, N_l is usually defined as the repetition of t blocks, usually two, composed by batch normalization, ReLU and convolution.

While *residual blocks* present a repetition of few blocks, which are sequentially connected, DenseNets extend this idea with another type of architecture. Dense blocks are characterized by a connectivity pattern that recursively concatenates all previous feature outputs. The output x_l of a DenseNet layer is defined as:

$$x_l := N_l([x_{l-1}, x_{l-2}, x_{l-3}, ..., x_0]), \tag{2}$$

where each layer is a composition of all previous ones through concatenation $[\cdots]$. The main characteristic of the densely connected block is the ability to explore information from previous outputs, given features are not shared across steps. Hence, different features may extract the same information multiple times, leading to a high redundancy block.

The residual network's main limitation resides in its element-wise summation operation for fusing information. This operation may squash useful features from preceding

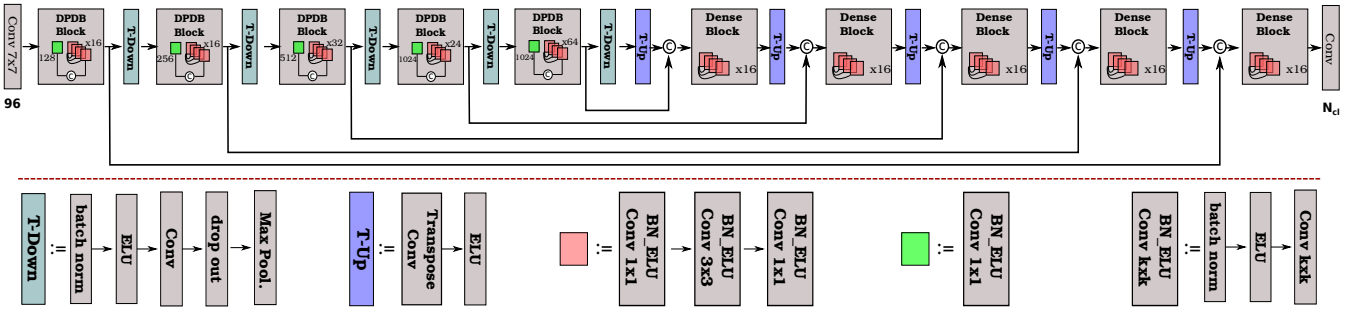

Fig. 2: DPDB-Net Architecture. Only convolutional, DPDB block, transition down, transition up and dense block layers are visualized. The network before the first transition up layer is considered the encoder part. Such part is constituted by DPDB blocks, while the rest of the network constitutes the decoder part and uses densely connected layers. Below the architecture we present a short description of each of the main building blocks of the proposed architecture.

layers. The squashing problem can be interpreted as follows: given two vectors of weights $\mathbf{w}_1 \triangleq [w_{11}, ..., w_{1n}]$ and $\mathbf{w}_2 \triangleq [w_{21}, ..., w_{2n}]$ and an element-wise aggregation function $f_{ag}(\mathbf{w}_1, \mathbf{w}_2) \triangleq \mathbf{w}_1 + \mathbf{w}_2$, thus, if $w_{1j} \gg w_{2j}$ then $f_{ag}(\mathbf{w}_1, \mathbf{w}_2) \sim \mathbf{w}_1 + \boldsymbol{\epsilon}$ then the importance of the low magnitude weights vanishes. Additionally, its high number of parameters, makes very deep residual networks intractable. DenseNets on the contrary can provide a better efficiency in term of parameter usage. On the other hand, dense blocks have an excessive parameter growth, due to the concatenation operation, that can greatly limit the width of DenseNets.

In the following section we will present the Dual-Path Dense-Block (DPDB) approach which can incorporate both characteristics into a single block.

B. DPDB Block

Based on the previous analysis, we propose a new dense block called Dual-Path Dense-Block. Our block is different from the dual path network [33], which also combines concepts from ResNet and DenseNet: we give similar weights to each of the sides and do not use a residual block as main block adding a thin densely connect path.

Given $x_{l,R}$ and $x_{l,D}$ as the outputs for the *l-th* layers of the residual path and dense path, we formulate the DPDB path block as:

$$r_l := x_{l,R} + x_{l,D}, \tag{3}$$

$$h_l := G_l(r_l), \tag{4}$$

where Equation 3 defines the path that adds both outputs and feeds them to the final transformation function in Equation 4. The transformation function $G_l(\cdot)$ is responsible for making the next mapping or prediction. Path fusion is done through concatenation, in order to avoid the feature squashing problem.

C. DPDB Networks

The proposed network consists of a downsampling part, called *encoder*, and an upsampling part called *decoder*. Figure 2 presents the proposed DPDB-Net architecture.

The *encoder* side is built by stacking DPDB blocks. Each block is composed by a dual path and the structure of each path is designed in the bottleneck style. The residual path is composed by a residual connection which is integrated to the dense structure with a set of 1×1 convolution layer followed by a 3×3 convolution, and ends with a 1×1 convolution layer. Each convolution layer described inside the micro-block is in fact a batch normalization, ELU activation function, and convolution. The output of the final 1×1 convolution is split into two parts, one that is added element-wise to the residual path, and the second which is concatenated to the densely connected path. Both streams are concatenated in the final stage to serve as input to the next block.

Every block in the *encoder* part is followed by a *transition down* operation to reduce the spatial dimensionality of the feature maps. The *transition down* block is composed of a batch normalization, ELU, convolution, dropout and a final 2×2 max pooling operation. We also explored stride for dimensionality reduction but max pooling showed better performance.

Our *decoder* part is exclusively composed by *dense blocks*. Usually, this part of FCN architectures holds a great number of parameters and DPDB blocks with the same size in the *encoder* and *decoder* part will make such architecture as intractable as Residual Nets. As presented in Section III-A *residual blocks* present the squashing problem when features of different magnitudes are fused. Such problem is potentially intensified in the *decoder*, given that skip connections provide features from distant convolution stages, therefore filtering out low magnitude features. Densely connected blocks perform concatenation operations, thus not suffering from the squashing problem and been more suitable for decoders.

Similarly to the *transition down* operation for the encoder part, we count with its counter-part for the up-sampling operation. Such module is called *transition up* and consists of a transposed convolution followed by an ELU activation function that upsamples the previous feature maps. The upsampled feature maps are concatenated to the corresponding skip connection from the encoder part. In order to overcome the feature map explosion of densely connected blocks, the input of such layer is not concatenated with its output. Therefore, we do not suffer from the linear growth in the

number of features of dense blocks, allowing us to build very deep architectures without feature map explosion.

Network	Output	DPDB Full
Input	360×480	-
Conv1	360×480	7×7, 96
DPDB_1	360×480	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 240 \end{bmatrix} \times 4$
DPDB_2	180×240	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 384 \end{bmatrix} \times 5$
DPDB_3	90×120	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 832 \end{bmatrix} \times 7$
DPDB_4	45×60	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 1336 \end{bmatrix} \times 10$
DPDB_5	22×30	$\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024 \\ 1 \times 1, 1984 \end{bmatrix} \times 12$
DB_1	22×30	$\begin{bmatrix} 1 \times 1, 912 \\ 3 \times 3, 16 \end{bmatrix} \times 12$
DB_2	45×60	$\begin{bmatrix} 1 \times 1, 480 \\ 3 \times 3, 16 \end{bmatrix} \times 10$
DB_3	90×120	$\begin{bmatrix} 1 \times 1, 320 \\ 3 \times 3, 16 \end{bmatrix} \times 7$
DB_4	180×240	$\begin{bmatrix} 1 \times 1, 208 \\ 3 \times 3, 16 \end{bmatrix} \times 5$
DB_5	360×480	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 16 \end{bmatrix} \times 4$
Conv Final	360×480	1×1, N_{cl}

TABLE I: DPDB architecture in more detail. For brevity *transition down* and *transition up* modules are not shown. The network has 217 convolutional layers. Most of them are in the *encoder* part (123 layers). The remaining layers are in the *decoder* (84 layers) and in the transitions with 5 layers each. In the table we use the following notation: DPDB stands for the Dual-Path Dense-Block module, DB for *Dense Block* and N_{cl} for the number of classes.

Table I provides more details on our DPDB-Net architecture. DPDB-Net has 217 convolutional layers, where the first convolution is a 7×7 layer in which we aim to obtain features with a larger field of view. The following 123 layers constitute its *encoder* part and the other 84 layers represent the decoding module. We count with 10 extra convolution layers in the transition down and up blocks. The final layer of the network is a 1×1 convolution followed by a softmax non-linearity to provide pixel-wise per-class distribution. One characteristic of the reduction of the feature map explosion problem is that we can provide a larger pre-softmax feature map.

IV. Experiments

We evaluated the performance of our network on two datasets, the Freiburg Forest dataset [31] and the CamVid dataset [4]. The implementation was based on the publicly available TensorFlow learning toolbox [1] and all experiments were carried on an NVIDIA Titan X GPU.

A. Dataset and Evaluation metrics

1) Datasets:

The Freiburg Forest dataset is an outdoor dataset for unstructured semantic segmentation. Unstructured semantic scene understanding plays an important role for robots operating in real world scenarios. The dataset was proposed by [31] and contains $15,000$, images which correspond to traversing about $4.7km$ each day. The dataset further contains 325 images with pixel-level groundtruth, which are divided into 203 images for training and 122 images for testing, with five classes, which are *sky*, *trail*, *grass*, *vegetation* and *obstacles*.

CamVid is a street scene understanding dataset which consists of five video sequences. The reported results use the same experimental setup proposed by [2], where 367 frames are training images, 100 images for validation and 233 images constitute the testing set. The images are 360×480 and the dataset differentiates between 11 semantic categories. One difference of CamVid to other datasets is its strong spatial relationship among different categories.

2) Evaluation Metrics: For the experiments we report the Mean IoU and the Global Avg. metrics. Mean IoU is the ratio of correct classified pixels in a class over the union set of pixels predicted to this class and groundtruth, and then averaged over all class $\frac{1}{N_{cl}} \sum_i \frac{t_{ii}}{T_i + \sum_j t_{ji} - t_{ii}}$. Global Avg. is the percentage of correctly classified pixels over the whole dataset $\frac{\sum_i t_{ii}}{\sum_i T_i}$. Given N_{cl} as the number of semantic classes and T_i the total number of pixels in class i, while t_{ij} indicates the number of pixels which belongs to class i and predicted to class j.

B. Architecture and training details

The network was trained in a single stage manner. DPDB-Net was trained using Adam solver [15], with an initial learning rate of 2×10^{-4} which decay $10\times$ every 2×10^5 iterations. All models are trained on data augmented images with multi-window random crop and vertical flip. We also apply mean subtraction to images and weight class balancing. We monitor the mean IoU every 100 iterations. We regularize

TABLE II: Results on Freiburg Forest dataset.

Method	Sky	Trail	Grass	Veg	mIoU
ParseNet [20]	87.78	81.82	85.20	85.20	85.0
M-Net [25]	89.26	82.41	84.93	88.70	86.3
Fast-Net [24]	90.46	84.51	86.72	**90.66**	88.0
GCN [26]	91.94	86.29	86.44	88.73	88.3
Ours - dense blocks	91.94	86.75	86.18	89.45	88.6
Ours - Full	**92.30**	**87.28**	**87.80**	90.14	**89.4**

TABLE III: Results on CamVid dataset.

Method	Pretrained	Temporal Inf.	Building	Tree	Sky	Car	Sign	Road	Pedestrian	Fence	Pole	Sidewalk	Cyclist	Global Avg.	mIoU
Superparsing [30]	✔	✗	70.4	54.8	83.5	43.3	25.4	83.4	11.6	18.3	5.2	57.4	8.9	–	42.0
ALE [17]	✔	✗	73.4	**70.2**	**91.1**	**64.2**	24.4	**91.1**	29.1	**31.0**	13.6	**72.4**	28.6	–	53.6
Liu [19]	✔	✗	66.8	66.6	90.1	62.9	21.4	85.8	28.0	17.8	8.3	63.5	8.5	–	47.2
SegNet [2]	✔	✗	68.7	52.0	87.0	58.5	13.4	86.2	25.3	17.9	16.0	60.5	24.8	62.5	46.2
DeconvNet [23]	✔	✗	–	–	–	–	–	–	–	–	–	–	–	85.9	48.9
FCN8s [21]	✔	✗	–	–	–	–	–	–	–	–	–	–	–	83.1	52.0
STFCN [10]	✔	✔	**73.5**	56.4	90.7	63.3	17.9	90.1	31.4	21.7	18.2	64.9	29.3	–	50.6
Reseg [32]	✔	✔	–	–	–	–	–	–	–	–	–	–	–	**88.7**	**58.8**
Ours - dense blocks	✗	✗	66.9	62.4	81.1	45.5	28.8	83.2	38.9	26.8	22.5	62.0	22.4	81.0	49.1
Ours - Full	✗	✗	72.4	62.9	88.6	61.9	**30.0**	88.8	**44.8**	26.1	**23.6**	69.4	**33.1**	**85.2**	**54.7**

our model with a weight decay of 10^{-4} and a dropout rate of 0.1.

C. Freiburg Forest Experiments

The Freiburg Forest dataset is a new benchmark for unstructured semantic segmentation. The dataset has 5 different classes, namely *sky*, *trail*, *grass*, *vegetation* and *obstacles*. While most of the dataset is fairly balanced, the class obstacle has a frequency of 0.85% and can be considered an outlier for comparison. Thus, for our experiments, we excluded this class.

Table II reports the obtained results and comparisons to the baseline. The experiments also points out the importance of the DPDB block in the encoder side of the proposed architecture. Even the recent proposed Global Convolution Network (GCN) [26] underperforms in comparison to our DPDB-Net. Additionally, we tested our architecture using only densely connected layers. We can notice that such architecture yields competitive results with GCN, however it is still not able to outperform to our complete approach.

Figure 3 shows qualitative results for our approach on the Freiburg Forest dataset. The obtained results depict the high mean IoU values presented previously. All examples are qualitatively similar to the groundtruth. However, small errors occur between vegetation and sky.

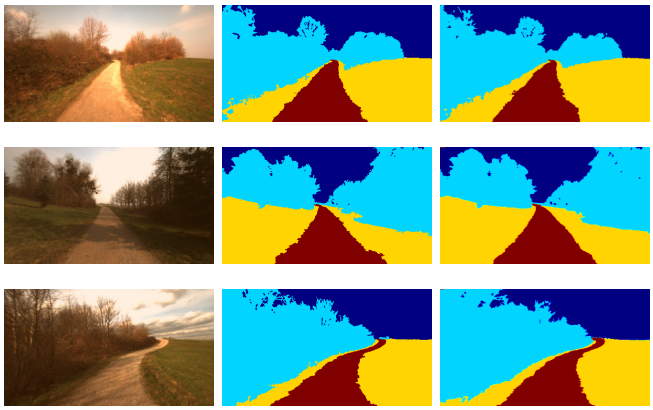

Fig. 3: Qualitative results on the Freiburg Forest test set. The rows represent from left to right: Input image, ground truth and prediction of the DPDB-Net.

D. CamVid Experiments

CamVid is a dataset of fully segmented videos for urban scene understanding [4]. Table III report our results with dense blocks only and with our full approach with DPDB blocks in the encoder part. The results show a clear superiority of the DPDP block, consistently improving the IoU for all classes and presenting a mean IoU gain superior to 5%. Our baseline ranges from traditional methods like [17], [19], [30] to fully convolutional approaches such as [2], [21], [23]. In addition to standard single frame approaches, we also compared our network with the Spatio Temporal FCN proposed at [10] and to a recurrent segmentation approach called Reseg [32].

When compared to other methods, we show that DPDB-Net achieves state-of-the-art results when only single frame approaches are taken into account. Particularly, we observe that the most challenging classes, such as *sign*, *pedestrian*, *pole* and *cyclist*, benefit notably from our approach. As the CamVid dataset corresponds to video frames it contains temporal information. Given the temporal consistency of this dataset, methods which incorporate spatio-temporal regularization to segmentation present clear advantage over single frame techniques. Reseg [32] is a method that employ recurrent layers to provide temporally consistent segmentation masks and constitutes the current overall state-of-the art approach. Our method is able to present competitive results even to spatio-temporal approaches, being even superior to STFCN [10], which is another FCN that incorporates temporal information to segmentation. No post-processing temporal regularization is used in the proposed architecture. However, we believe such addition would yield additional improvements. Unlike all the compared approaches, that make use of a pre-trained *encoder* on large datasets like ImageNet [8] our DPDB encoder have not been pre-trained but we could likely benefit from training on these datasets.

Figure 4 shows qualitative segmentation masks obtained by our approach on the CamVid dataset. They ratify our quantitative results, showing sharp and well defined segmen-

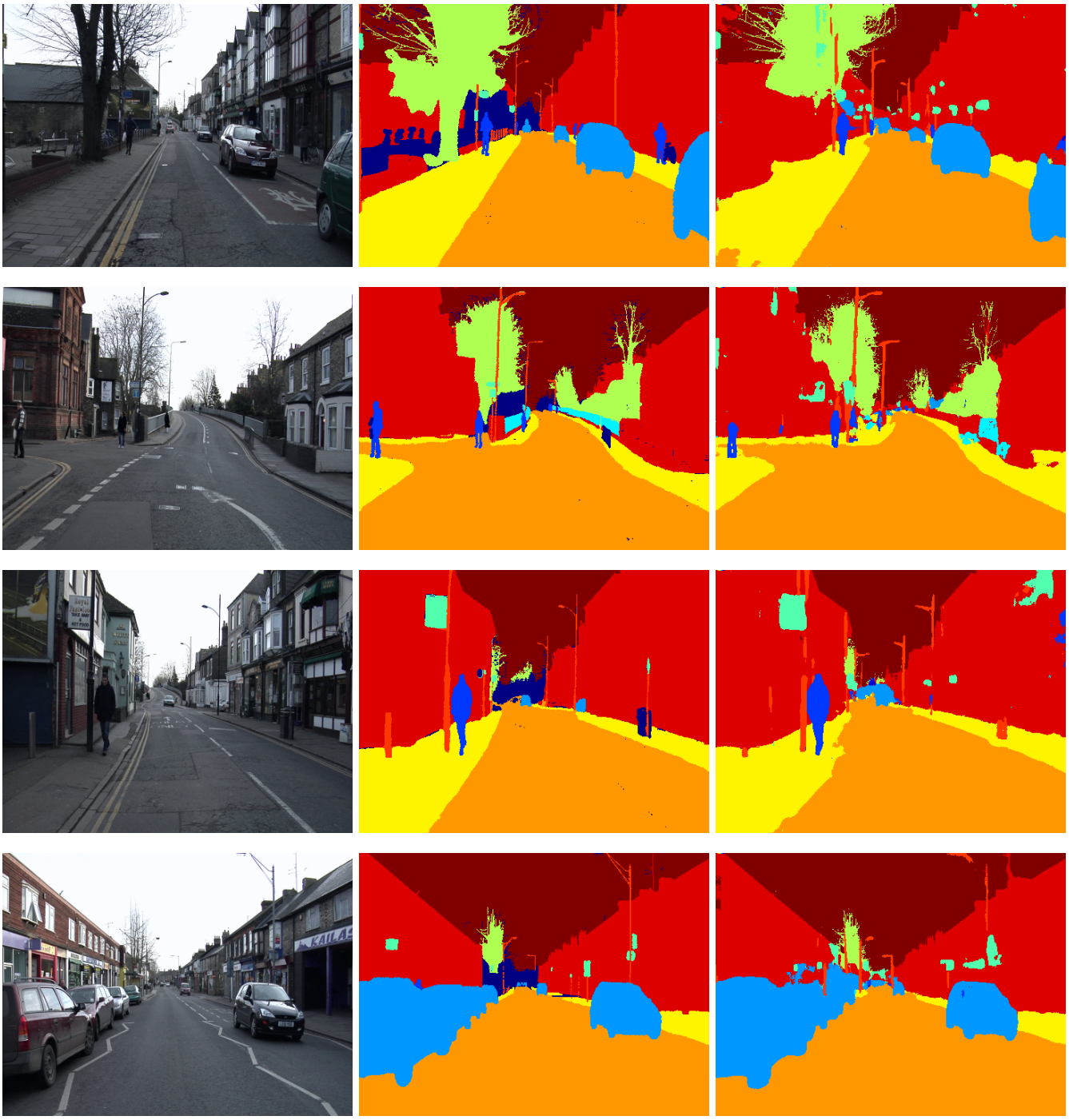

Fig. 4: Qualitative results on the Camvid test set. The rows represent from left to right: Input image, ground truth and prediction of the DPDB-Net. The first row shows examples in which the segmentation approach performs accurately, however the tree class is not completely segmented. In the second and third rows, it can be seen one of the strongest characteristics of our architecture, which is highly detailed segmentation to challenging classes, such as person, pole and sign. The last row presents an example with less fine structures. The last two rows also depicts common prediction mistakes, such as between sign and building, which can be noticed in the third row, top-right side and in the fourth row, top-left side.

tation masks for fine structures. Classes like *pole*, *pedestrian* and *sign* appear highly detailed. Some errors are related to problems like transitions between classes in small areas and class confusions like between fence and building, which can be seen in the second row (top-to-bottom). Another common confusion is between *sign* and *building*, which can be noticed in the third row, top-right side and in the fourth row, top-left side.

V. Conclusions

In this paper, we presented a new architecture called Dual-Path Dense-Block Network (DPDB-Net). We introduced a dense block that incorporates feature re-usage and new feature exploration in the encoder of fully convolutional networks. The resulting DPDB-Net is an architecture with 217 layers. It improves the state-of-the-art performance on a challenging unstructured semantic segmentation dataset (Freiburg Forest) and presents competitive results on the CamVid dataset without requiring post-processing, pre-training, or temporal regularization modules.

References

[1] M. Abadi, A. Agarwal, P. Barham, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

[3] Jonathan T Barron and Ben Poole. The fast bilateral solver. *ECCV*, 2016.

[4] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 2008.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016.

[7] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units. In *ICLR*, 2016.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[10] Mohsen Fayyaz, Mohammad Hajizadeh Saffar, Mohammad Sabokrou, Mahmood Fathy, and Reinhard Klette. STFCN: spatio-temporal FCN for semantic video segmentation. *CoRR*, abs/1608.05971, 2016.

[11] Xiang Gao and Tao Zhang. Unsupervised learning to detect loops using deep neural networks for visual slam system. *Autonomous Robots*, 41(1):1–18, 12 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[13] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[14] Varun Jampani, Martin Kiefel, and Peter V. Gehler. Learning sparse high dimensional filters: Image filtering, dense crfs and bilateral neural networks. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012.

[17] Lubor Ladicky, Christopher Russell, Pushmeet Kohli, and Philip H. S. Torr. Associative hierarchical crfs for object class image segmentation. In *ICCV*, pages 739–746, 2009.

[18] G. Lin, C. Shen, A. van dan Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 2016.

[19] Buyu Liu and Xuming He. Multiclass semantic video segmentation with object-level active inference. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[20] Wei Liu, Andrew Rabinovich, and Alexander Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, nov 2015.

[22] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3376–3385, June 2015.

[23] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015.

[24] G. L. Oliveira, W. Burgard, and T. Brox. Efficient deep models for monocular road segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[25] Gabriel L Oliveira, Claas Bollen, Wolfram Burgard, and Thomas Brox. Efficient and robust deep networks for semantic segmentation. *The International Journal of Robotics Research*, 2017.

[26] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters – improve semantic segmentation by global convolutional network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.

[29] Niko Suenderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[30] Joseph Tighe and Svetlana Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, pages 352–365, 2010.

[31] Abhinav Valada, Gabriel L. Oliveira, Thomas Brox, and Wolfram Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. *2016 International Symposium on Experimental Robotics*, pages 465–477, 2017.

[32] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[33] Huaxin Xiao Xiaojie Jin Shuicheng Yan Jiashi Feng Yunpeng Chen, Jianan Li. Dual path networks. *arXiv preprint arXiv:1707.01629*, 2017.