DCNAS: Densely Connected Neural Architecture Search for Semantic Image Segmentation Supplementary Materials

Xiong Zhang[1], Hongmin Xu[2], Hong Mo[3], Jianchao Tan[2], Cheng Yang[1], Lei Wang[1], Wenqi Ren[4]
[1]Joyy Inc., [2] AI Platform, Kwai Inc., [3]State Key Lab of VR, Beihang University, [4]SKLOIS, IIE,CAS

1. Methodology

1.1. Shape Alignment Layer

The shape-alignment layer is in a multi-branch parallel form, which aims at transforming the semantic features (with various shapes, including spatial resolutions and channel widths) to the target shape. In practice, we use bilinear-upsampling or 3×3 convolutional layer with proper stride to adjust the spatial resolution and insert 1×1 convolutional layer to perform channel alignment.

1.2. Stem and Head Module

In order to deliver end-to-end searching and training, we hand-craft a stem module that aims at extracting feature-pyramids for DCNAS, and we also design a simple prediction head for aggregating all the feature-maps that from DCNAS to make the final prediction. In brief, the stem block consists of a 7×7 convolution and four 3×3 convolutions with stride 2, the 7×7 convolution has 32 filters, while the last four 3×3 convolutions have $F, 2F, 4F, 8F$ filters, respectively. Regarding the structure of the prediction head, we concatenate the (upsampled) semantic features from $\{O_{(1/4,L)}, O_{(1/8,L)}, O_{(1/16,L)}, O_{(1/32,L)}\}$ then apply an extra 3×3 convolution and a 1×1 convolution to make the final prediction.

1.3. Decoding Algorithm

Once the searching procedure terminates, one may derive the suitable operator for each mixture layer and the optimal architecture based on the architecture parameters α and β. For mixture layer $\ell_{(s,l)}$, we select the candidate operation that has maximum operation weight, *i.e.,* $\arg\max_{o \in \mathcal{O}} \alpha^{o}_{(s,l)}$. Regrading the network architecture, we utilize a Breadth-First Search algorithm to derive the network architecture in a back to front order, and the algorithm is listed as bellow:

Algorithm 1: Decoding Network Structure

```
Data: Architecture parameters β
Result: Derived optimal connections opt_conns
opt_conns := {};
int_nodes := {(1/4, L), (1/8, L), (1/16, L), (1/32, L)};
while int_nodes not empty do
    let (s, l) = int_nodes.pop();
    foreach candidate connection (s', l') → (s, l) do
        if β_{(s',l')→(s,l)} ≥ 0 then
            if l' > 0 then
                int_nodes.append((s', l'));
            if (s', l') → (s, l) ∉ opt_conns then
                opt_conns.append((s', l') → (s, l))
return opt_conns
```

1.4. Regularization Terms

In practice, to obtain a faster convergence, we introduce several instrumental regularization terms as bellow,

1. Since we may select the optimal operator $\arg\max_{o \in \mathcal{O}} \alpha^o_{(s,l)}$, we introduce the entropy regularization term

$$\mathcal{L}_\alpha = - \sum_{s \in \mathbb{S}, l \leq L} \underbrace{\sum_{o \in \mathcal{O}} w^o_{(s,l)} \ln w^o_{(s,l)}}_{\text{to derive a one-hot distribution over } \mathcal{O}} \tag{1}$$

over architecture parameters α to derive a one-hot distribution of the operators in each mixture layer $\ell_{(s,l)}$.

2. The insignificant transmissions always affect the aggregating operation, which will affect the convergence. Thus we introduce a regularization term as:

$$\begin{aligned} \mathcal{L}_\beta &= \sum_{s' \in \mathbb{S}, s \in \mathbb{S}, l \leq L, l' < l} - \underbrace{\frac{1}{(1 + \exp^{-\beta_{(s',l') \to (s,l)}})}}_{\text{importance of connection} (s',l') \to (s,l)} \cdot \ln \frac{1}{(1 + \exp^{-\beta_{(s',l') \to (s,l)}})} \\ &= \sum_{s' \in \mathbb{S}, s \in \mathbb{S}, l \leq L, l' < l} \ln(1 + \exp^{-\beta_{(s',l') \to (s,l)}}) / (1 + \exp^{-\beta_{(s',l') \to (s,l)}}) \end{aligned} \tag{2}$$

to help solve this challenge.

3. We also want to constrain the minimum and maximum number of fusion modules that connect to it to be 1 and k, we choose to apply the Lagrangian multiplier method to reformulate such constraints as one regularization term:

$$\begin{aligned} \mathcal{L}_{con} = &\sum_{s \in \mathbb{S}, l \leq L} \max(1 - \underbrace{\sum_{s' \in \mathbb{S}, l' < l} \frac{1}{1 + \exp^{-\beta_{(s',l') \to (s,l)}}}}_{\text{in degree}}, 0) \\ &+ \sum_{s \in \mathbb{S}, l \leq L} \max(\underbrace{\sum_{s' \in \mathbb{S}, l' < l} \frac{1}{1 + \exp^{-\beta_{(s',l') \to (s,l)}}}}_{\text{in degree}} - k, 0). \end{aligned} \tag{3}$$

1.5. Searching Protocols

We conduct architecture search on the Cityscapes dataset [4], more specifically, we divide the training samples with fine annotations into two parts, with $2,000$ and 975 samples respectively. The $2,000$ training samples are used for updating convolutional weights w while the 975 samples are used for optimizing architecture parameters $\{\alpha, \beta\}$. Besides we make use of the 500 fine annotated validation samples for model selection. For searching configurations, we set the initial learning rate as 0.01 and 0.0005 for w and $\{\alpha, \beta\}$ respectively, schedule the learning rate with polynomial policy with factor $(1 - (\frac{iter}{iter_{max}})^{0.9})$, apply weight decay for w with coefficients 0.0001. Following conventional data augmentation paradigm, we resize the image with random scale $[0.5, 2.0]$, then random crop a patch with size 1024×512, apart from that, we do not employ any augmentation tricks though which shall further improve the performance. Consider that the architecture parameters are hard to optimize, we exploit the Adam optimizer to update $\{\alpha, \beta\}$, while for convolutional weights w, we adopt the general SGD for sake of better convergence. The searching procedure takes 1.4 days for 120 epochs on 4 GPUs.

1.6. Training Protocols

We evaluate our optimal model on Cityscapes [4], PASCAL VOC 2012 [6], ADE20K [25], and PASCAL-Context [17] datasets. We share the same training protocols for all the datasets mentioned above, except for particular dataset-specific configurations. Specifically, during training we set the initial learning rate to 0.01 and adopt the polynomial scheduler [16] to drop the learning rate, we exploit the syncBN [19, 15] to synchronize the mean and variance across different GPUs, we conduct the experiments with batch size 32 on 4 GPUs, and we use the SGD with momentum 0.7 to optimize the convolutional weights. Data are augmented by random scaling in range of [0.5, 2.0], random horizontal flipping, and random cropping, where we set the crop size 1024×512 for Cityscapes and 513×513 for others. Considering the complexity and scale of the benchmarks, empirically, we train the model on tough Cityscape for 800 epochs, concerning other benchmarks, we set the training epochs to 240.

Methods	#Params	FLOPs	FPS
Auto-DeepLab	44.42M	695.03G	-
DCNAS (Ours)	21.49M	294.57G	6.3

Table 1. **Comparison with Auto-DeepLab.** FPS is evaluated on 1080Ti GPU with CUDA8.0 (inputs with 2048 x 1024).

1.7. Correlation of Performance

We utilize the Pearson Correlation Coefficient (ρ) and Kendall Rank Correlation (τ) Coefficient to quantify the correlation of performance between searching and training situations. Denote (X, Y) is a two dimension random variables, and $\{x_i, y_i\}_{i=1}^n$ are n observations, the we may estimate the Pearson Correlation Coefficient (ρ) as,

$$\begin{aligned} \rho &= \frac{COV(X,Y)}{\sqrt{D(X)D(Y)}} \\ &= \frac{E(XY) - E(X)E(Y)}{\sqrt{D(X)D(Y)}} \end{aligned} \tag{4}$$

and we may calculate the Kendall Rank Correlation (τ) Coefficient with,

$$\tau = \frac{\sum_{i \leq n, j<i} \mathbb{1}\{x_i<x_j\}\mathbb{1}\{y_i<y_j\} - \sum_{i \leq n, j<i} \mathbb{1}\{x_i<x_j\}\mathbb{1}\{y_i>y_j\}}{\begin{pmatrix} n \\ 2 \end{pmatrix}} \tag{5}$$

2. Experiments

2.1. Model Efficiency

Since we do not guide the searching process to derive a sparse and compact model, therefore, the derived model tends to uses all nodes, and searching for sparse while compact structures for realtime semantic image segmentation is indeed our further work. We also compare our optimal model with contemporary work of [13, 3], as Table 1 presents the comparison results, our method achieves better performance with lower FLOPs.

2.2. Qualitative Results

To better understand the capability of DCNAS, we drawn several examples from the validation set (Figure 1) and the testing part (Figure 2). The DCNAS can produce precise predictions about small targets (*e.g.,* tsign, pole) in the scene thanks to the ability to capture subtle details in high-resolution imageries. Meanwhile, owing to the capacity to extract long-range global information, DCNAS is good at estimating the segmentation mask of massive while complicated objects (*e.g.,* bus, truck, terrain, sidewalk, rider, building).

2.3. Ablation Study

To better understand the impact of different design choices, we evaluate our framework in various settings. Two main design choices exist in this work, the impact of hyper-parameters L and k, and the effect of the novel regularization terms. We still take the mIoU as the evaluation metric for searching (S-mIoU) and training (T-mIoU) periods on the validation dataset of the Cityscapes.

Operator Space $\mathcal{O}$. In our work, we use a totally different operator space compared with Auto-DeepLab [13]. To make an apple to apple comparison, we evaluate the performance of DCNAS with the same operator space in [13], and Table 2 reveals that MobileNetV3 operator space do not substantially affect both the performances and the correlation coefficients.

Hyperparameter k. k is an important parameter in regularization term $\mathcal{L}_{con}$, as presents in In Table 6, as k increases, the performance grows first and then decreases. Because, (1) small k will reduce the search space largely, which may miss the optimal architecture and results in poor performance, (2) large k will lead a tremendous search space, from which explore promising architecture is challenging. As a compromise, we choose $k = 3$ in our experiments.

Regularization Terms. Regarding the regularization terms, we observe that the $\mathcal{L}_\beta$ is crucial to performance improvement, and adding the regularization term $\mathcal{L}_{con}$ can yield marginal performance improvement. We think there are two reasons:

Methods	MobileNetV3	ρ	τ	S-mIoU	T-mIoU
Auto-DeepLab	✗	0.31	0.21	35.1	80.3
Auto-DeepLab	✓	0.34	0.24	35.0	80.4
DCNAS (Ours)	✗	0.74	0.53	69.7	81.0
DCNAS (Ours)	✓	0.73	0.55	69.9	81.2

Table 2. **Ablation Study.** The table investigate the impact ofs MobileNetV3 operator space.

Budget	Regularizers	S-mIoU	T-mIoU
$1\times$	✓	69.9	81.2
$1\times$	✗	48.1	76.5
$2\times$	✗	60.2	78.9
$3\times$	✗	67.5	80.7
$4\times$	✗	70.0	81.4

Table 3. **Ablation Study.** We explore the impact of regularizers over performance under several searching budgets (GPU Days).

$\mathcal{L}_{con}$	$\mathcal{L}_{\beta}$	$\mathcal{L}_{\alpha}$	S-mIoU	T-mIoU
✗	✗	✗	48.1	76.5
✗	✗	✓	51.3	77.1
✗	✓	✗	63.9	79.4
✗	✓	✓	65.3	79.9
✓	✗	✗	55.2	78.3
✓	✗	✓	58.7	78.4
✓	✓	✗	69.6	80.9
✓	✓	✓	69.9	81.2

Table 4. **Ablation Study.** We investigate the impact of different regularization terms by comparing the performance on Cityscapes validation set, in which L and k are set to be 14 and 3.

L	r	GPU Days	S-mIoU	T-mIoU
8	1	8.6	61.4	73.1
8	$1/2$	5.2	60.9	73.0
8	$1/4$	2.8	60.1	72.8
8	$1/8$	1.5	57.9	71.1
8	$1/16$	0.9	55.2	68.3
14	1	16.2	71.1	81.31
14	$1/2$	9.8	70.8	81.3
14	$1/4$	5.6	69.9	81.2
14	$1/8$	3.0	67.6	80.6
14	$1/16$	1.9	63.8	78.9

Table 5. **Ablation of Sampling Ratio.** The table presents the impact of sampling ratio r in mixture layer.

L	k	S-mIoU	T-mIoU
14	1	59.4	76.2
14	2	66.8	79.5
14	3	**69.7**	**81.2**
14	4	69.6	81.0
14	5	69.3	80.9

Table 6. **Ablation of k.** We present the performance of our DCNAS on Cityscapes validation dataset with varying configurations of k.

(1) the regularization term $\mathcal{L}_{\beta}$ forces the relaxed continuous representations of path connection to be 0 or 1, which stabilizes the network architecture; (2) constraint $\mathcal{L}_{con}$ helps prune insignificant path-level connections and derive a sparse model struc-

Methods	higher is better			lower is better		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	AbsRel	RMSE	log10
Make3D [21]	0.447	0.745	0.897	0.349	1.214	-
Joint HCRF [22]	0.605	0.890	0.970	0.220	0.824	-
Liu *et al.* [14]	0.650	0.906	0.976	0.213	0.759	0.087
Eigen *et al.* [5]	0.769	0.950	0.988	0.158	0.641	-
Li *et al.* [12]	0.789	0.955	0.988	0.152	0.611	0.064
Gur *et al.* [8]	0.797	0.951	0.987	0.149	0.546	0.063
Chakrabarti *et al.* [1]	0.806	0.958	0.987	0.149	0.620	-
Laina *et al.* [9]	0.811	0.953	0.988	0.127	0.573	0.055
Xu *et al.* [23]	0.811	0.954	0.987	0.121	0.586	0.052
Lee *at al.* [10]	0.815	0.963	0.991	0.139	0.572	-
DORN [7]	0.828	0.965	0.992	0.115	0.509	0.051
Geonet [20]	0.834	0.960	0.990	0.128	0.569	0.057
Lee *et al.* [11]	0.837	0.971	0.994	-	0.538	-
Yin *et al.* [24]	0.875	0.976	0.994	0.108	**0.416**	**0.048**
Freeform [2]	**0.930**	**0.990**	**0.999**	**0.087**	0.433	0.052
Ours (DCNAS)	0.836	0.968	0.994	0.113	0.497	0.052

Table 7. **Quantitative performance on NYU Depth v2.** Our method achieves competitive result against state-of-the-art approaches. Though not state-of-the-art, we argue it is reasonable when considering the scale of NYU benchmark and the saturated performance.

ture, which further stabilizes the network architecture. Table 3 exhibits that those regularization terms is capable of reducing the searching cost substantially.

Moreover, as presented in Table 4, adding the regularization term $\mathcal{L}_\alpha$ can further improve the performance, since $\mathcal{L}_\alpha$ leads to a discrete distribution of operator space in each mixture-layer, which yields more stable and reliable training.

Sampling Ratio r in Mixture Layer. We also investigate the impact of the sampling ratio r in mixture layer. As Table 5 demonstrates that $1/4$ is a good trade-off between search efficiency and model accuracy.

2.4. Generalization Capability

To evaluate the generalization performance of our approach, we evaluate the DCNAS on task of monocular depth estimation on NYU Depth v2 [18], as Table 7 and Figure 3 present the quantitative and qualitative results. It is exciting that the quantitative and qualitative experiment reveals our DCNAS can generalize well to other dense image prediction task.

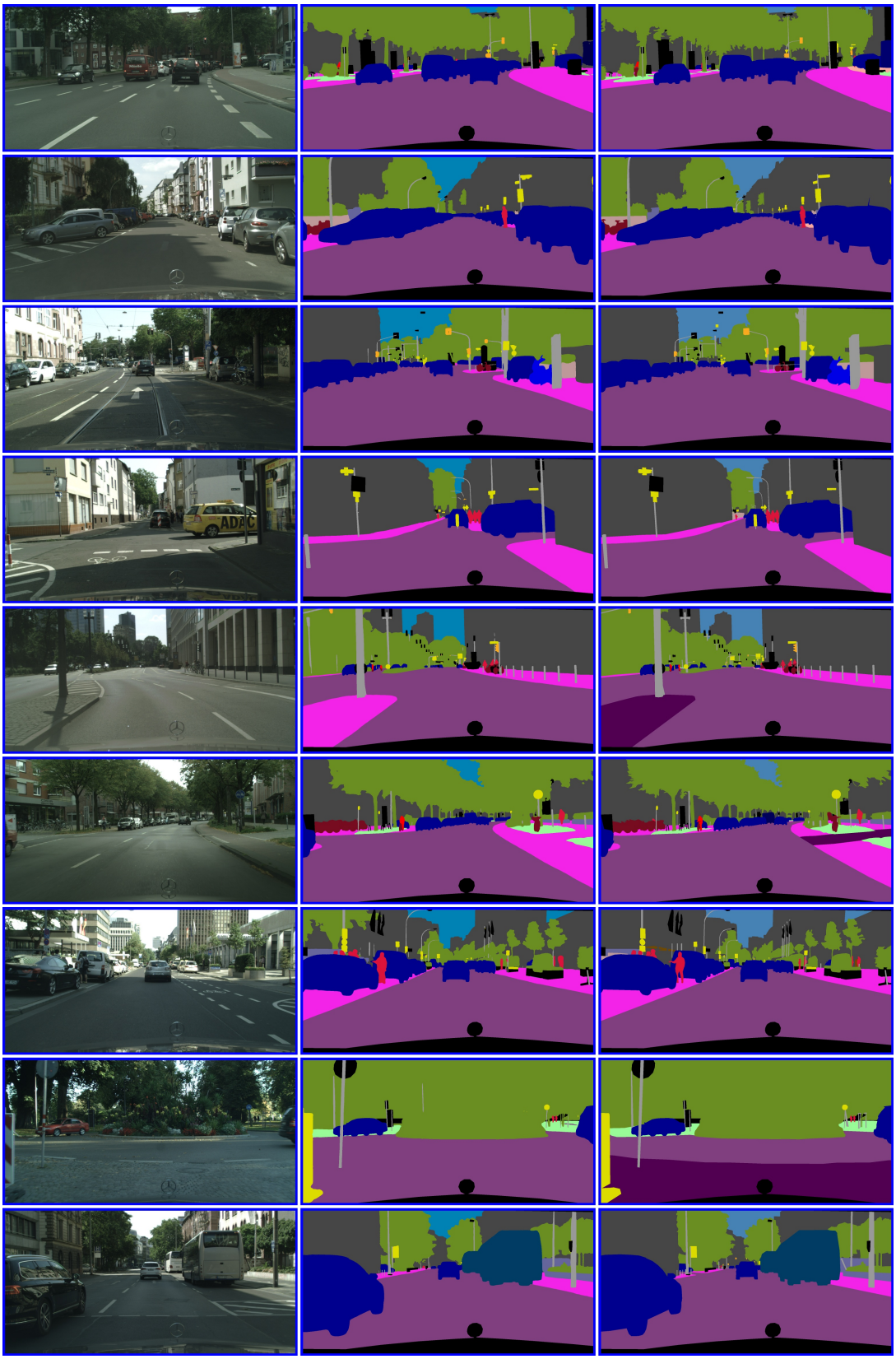

Figure 1. **Qualitatively results on cityscapes validation part.** We present several prediction results on Cityscapes validation part. The three columns represent the RGB input, prediction given by DCNAS, and the ground-truth, respectively.

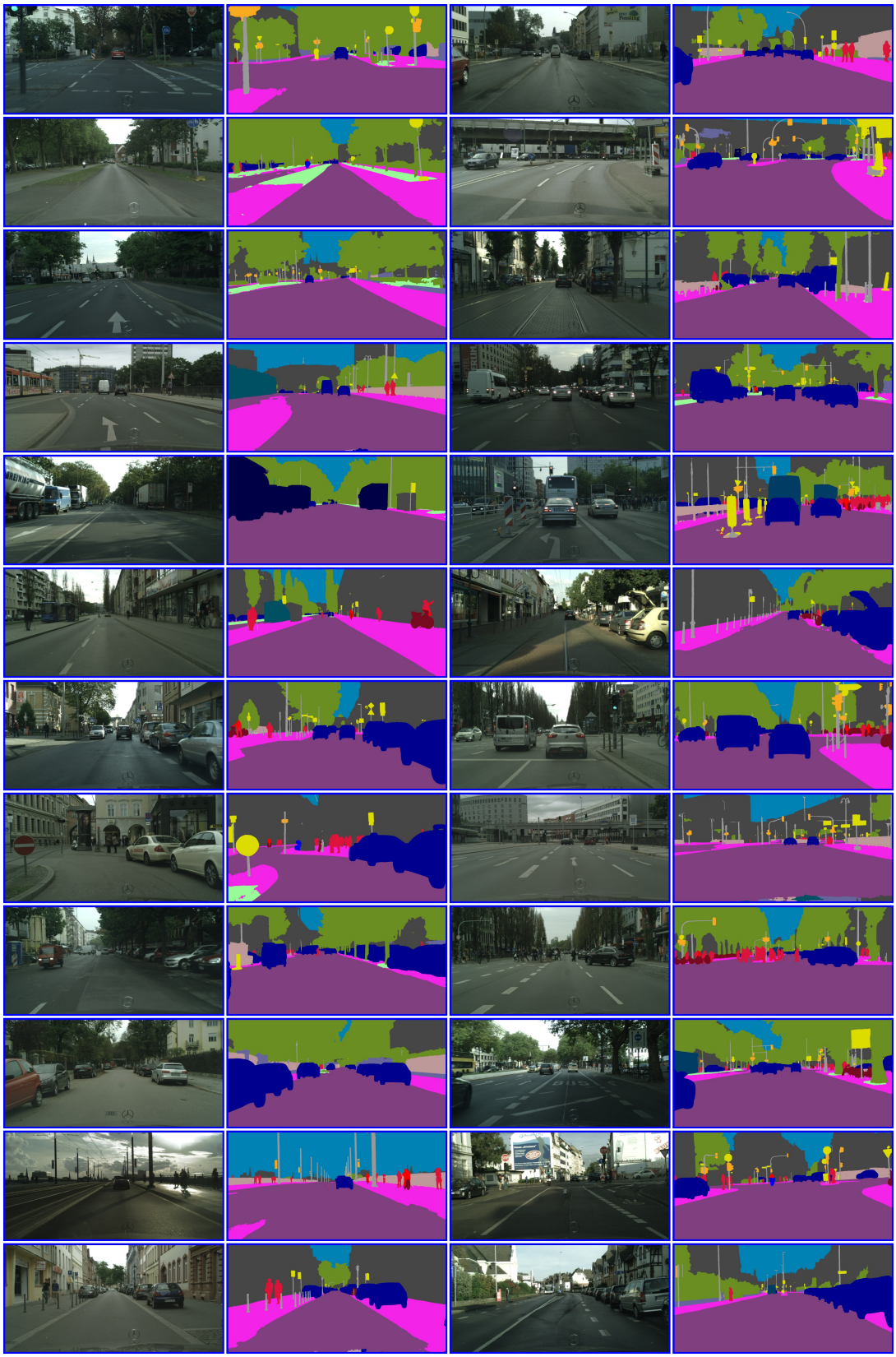

Figure 2. **Qualitatively results on Cityscapes test part.** We present several prediction results on Cityscapes test part. In which each row comprise two samples, each sample contains the RGB input and the result produced by DCNAS.

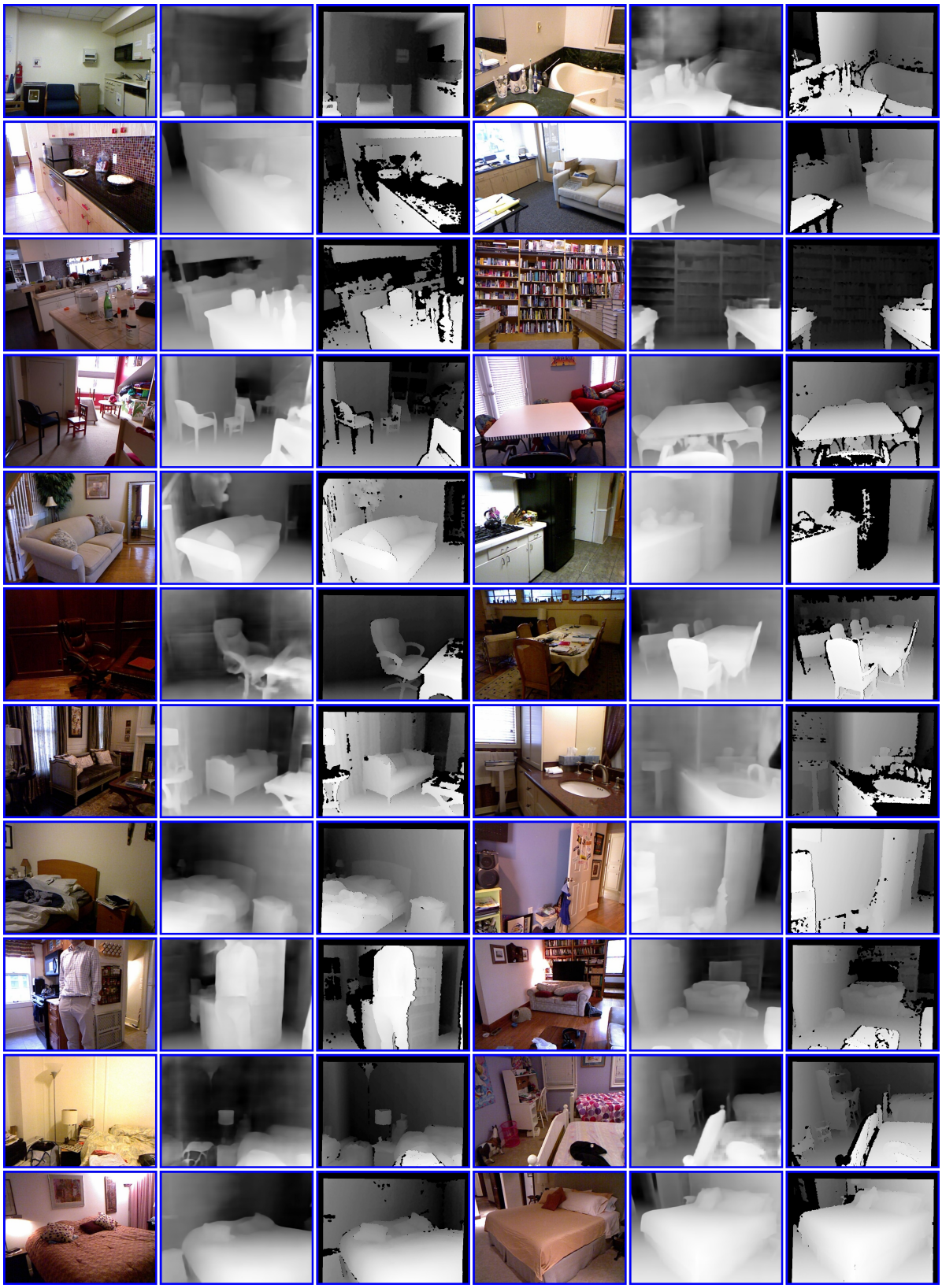

Figure 3. **Qualitatively results on NYU Depth v2.** We present several prediction results on NYU test part. In which each row comprise two samples, each sample contains the RGB input, the mask result produced by DCNAS, and the ground truth, respectively.

References

[1] Ayan Chakrabarti, Jingyu Shao, and Greg Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. In *Advances in neural information processing systems (NIPS)*, 2016. 5

[2] Julie Chang and Gordon Wetzstein. Deep optics for monocular depth estimation and 3d object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[3] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *Advances in neural information processing systems (NIPS)*, 2018. 3

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016. 2

[5] David Eigen and Rob 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 (ICCV)*, 2015. 5

[6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision (IJCV)*, 2010. 2

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 5

[8] Shir Gur and Lior Wolf. Single image depth estimation trained via depth from defocus cues. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019. 5

[9] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, 2016. 5

[10] Jae-Han Lee, Minhyeok Heo, Kyung-Rae Kim, and Chang-Su Kim. Single-image depth estimation based on fourier domain analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 5

[11] Jae-Han Lee and Chang-Su Kim. Monocular depth estimation using relative depth maps. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019. 5

[12] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 5

[13] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2019. 3

[14] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2015. 5

[15] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 2

[16] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 2

[17] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. 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 (CVPR)*, 2014. 2

[18] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European conference on computer vision (ECCV)*, 2012. 5

[19] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 2

[20] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018. 5

[21] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2008. 5

[22] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2015. 5

[23] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017. 5

[24] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 5

[25] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2017. 2