Disentangled Non-local Neural Networks

Minghao Yin[1⋆], Zhuliang Yao[1,2⋆], Yue Cao[2], Xiu Li[1], Zheng Zhang[2], Stephen Lin[2], and Han Hu[2]

[1] Tsinghua University
{yinmh17,yzl17}@mails.tsinghua.edu.cn li.xiu@sz.tsinghua.edu.cn
[2] Microsoft Research Asia
{yuecao,zhez,stevelin,hanhu}@microsoft.com

Table 1. Results with more NL and DNL blocks based on Mask R-CNN, using R50 as backbone with FPN, for object detection and instance segmentation on COCO 2017 validation set

	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{mask}	AP^{mask}_{50}	AP^{mask}_{75}
baseline	38.8	59.3	42.5	35.1	56.2	37.9
NL (c4 one)	39.6	60.3	43.2	35.8	57.1	38.5
NL (c5 all)	40.0	62.1	43.5	36.1	58.6	38.6
NL (c4c5 all)	40.1	62.3	43.5	36.0	58.9	38.3
DNL (c4 one)	40.3	61.2	44.1	36.4	58.0	39.1
DNL (c5 all)	41.2	62.7	44.7	37.0	59.5	39.5
DNL (c4c5 all)	41.4	63.2	45.3	37.3	59.8	39.8

1 More NL/DNL blocks for COCO Object Detection

In section 5.2 of the main paper, we follow the settings in [8] where 1 non-local (NL) or disentangled non-local (DNL) block is inserted right before the last residual block of c4. In this section, we investigate the performance of NL and DNL when more attention blocks are inserted into the backbone, as shown in Table 1.

While the proposed DNL method outperforms NL method by 0.7% bbox mAP and 0.6% mask mAP when 1 attention block is inserted into the backbone (denoted as "c4 one"), the gains brought by the proposed DNL method over the NL method are enlarged to 1.2% bbox mAP and 0.9% mask mAP, respectively, when every residual block of stage c5 is followed by 1 attention block (denoted as "c4 all"). The gains are further enlarged to 1.3% bbox mAP and 1.3% mask mAP when additionally every residual block of stage c4 is followed by 1 attention block (denoted as "c4 c5 all"). These results indicate that the DNL method can benefit more from increasing block number than the NL method.

⋆ Equal contribution. This work is done when Minghao Yin and Zhuliang Yao are interns at MSRA.

2 Detailed Proof of Proposition 1

The object function $O(\alpha, \beta)$ in Eq. (3) of the main paper can be rewritten as

$$\begin{aligned} O(\alpha,\beta) = &\sum_{i\in\Omega}(\mathbf{q}_i-\alpha)^T A(\mathbf{q}_i-\alpha) + \sum_{m\in\Omega}(\mathbf{k}_m-\beta)^T B(\mathbf{k}_m-\beta) \\ &- \sum_{i\in\Omega}\Big((\mathbf{q}_i-\alpha)^T(\mathbf{q}_i-\alpha)\Big) - \sum_{m\in\Omega}\Big((\mathbf{k}_m-\beta)^T(\mathbf{k}_m-\beta)\Big) \end{aligned} \tag{1}$$

where

$$A = \frac{\sum_{m,n\in\Omega}(\mathbf{k}_m-\mathbf{k}_n)(\mathbf{k}_m-\mathbf{k}_n)^T}{\sum_{m,n\in\Omega}(\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_m-\mathbf{k}_n)} \qquad B = \frac{\sum_{i,j\in\Omega}(\mathbf{q}_i-\mathbf{q}_j)(\mathbf{q}_i-\mathbf{q}_j)^T}{\sum_{i,j\in\Omega}(\mathbf{q}_i-\mathbf{q}_j)^T(\mathbf{q}_i-\mathbf{q}_j)} \tag{2}$$

We first prove that all eigenvalues of matrix A and B are smaller or equal than 1. Denote the eigenvalues of matrix A as $\lambda_1, ..., \lambda_d$. According to Cauchy–Schwarz inequality, we have

$$\begin{aligned} &\sum_{1\leqslant i\leqslant d}\lambda_i^2 = tr(A^TA) \\ &= \mathrm{tr}\left(\frac{\sum_{m,n\in\Omega}(\mathbf{k}_m-\mathbf{k}_n)(\mathbf{k}_m-\mathbf{k}_n)^T\cdot\sum_{s,t\in\Omega}(\mathbf{k}_s-\mathbf{k}_t)(\mathbf{k}_s-\mathbf{k}_t)^T}{\sum_{m,n\in\Omega}(\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_m-\mathbf{k}_n)\cdot\sum_{s,t\in\Omega}(\mathbf{k}_s-\mathbf{k}_t^T)(\mathbf{k}_s-\mathbf{k}_t)}\right) \\ &= \frac{\sum_{m,n,s,t\in\Omega}(\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_s-\mathbf{k}_t)\cdot tr\left((\mathbf{k}_m-\mathbf{k}_n)(\mathbf{k}_s-\mathbf{k}_t)^T\right)}{\left(\sum_{m,n\in\Omega}(\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_m-\mathbf{k}_n)\right)^2} \\ &= \frac{\sum_{m,n,s,t\in\Omega}\left((\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_s-\mathbf{k}_t)\right)^2}{\left(\sum_{m,n\in\Omega}(\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_m-\mathbf{k}_n)\right)^2} \quad \leq 1 \end{aligned} \tag{3}$$

Given Eq. (3), we have: $\forall 1 \leq i \leq d$, $\lambda_i \leq 1$. Similarly, we can prove all eigenvalues of matrix B are smaller or equal than 1. The hessian matrix of Eq. (1) with respect to α and β are non-positive definite matrix. The optimalα^* and β^* are thus the solutions of the following equations: $\frac{\partial O}{\partial \alpha} = 0$, $\frac{\partial O}{\partial \beta} = 0$.

For α^*, we have

$$\begin{aligned} &\frac{\partial O}{\partial \alpha^*} = \sum_{i=1}^{N_p} 2\left(\frac{\sum_{m,n}(\mathbf{k}_m-\mathbf{k}_n)(\mathbf{k}_m-\mathbf{k}_n)^T}{\sum_{m,n}(\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_m-\mathbf{k}_n)} - 1\right)(\mathbf{q}_i-\alpha^*) = 0, \\ &\Leftrightarrow \left(\frac{\sum_{m,n}(\mathbf{k}_m-\mathbf{k}_n)(\mathbf{k}_m-\mathbf{k}_n)^T}{\sum_{m,n}(\mathbf{k}_m-\mathbf{k}_n)^T(\mathbf{k}_m-\mathbf{k}_n)} - 1\right)\sum_{i=1}^{N_p} 2(\mathbf{q}_i-\alpha^*) = 0. \end{aligned} \tag{4}$$

To satisfy Eqn. 4, we have:

$$\sum_{i=1}^{N_p}(\mathbf{q}_i-\alpha^*) = 0. \tag{5}$$

The optimal α^* is thus

$$\alpha^* = \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{q}_i. \tag{6}$$

Similarly, the optimal β^* is computed as

$$\beta^* = \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{k}_i. \tag{7}$$

3 Proof for Eqn. 4 in the main paper

Here, we provide a proof for Eqn. 4 in Section 3.1. The dot product of query $\mathbf{q}_i$ and key $\mathbf{k}_j$ can be split into several terms by introducing a whitening operation on the key and query:

$$\mathbf{q}_i^T\mathbf{k}_j = \left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_j-\boldsymbol{\mu}_k\right)+\boldsymbol{\mu}_q^T\mathbf{k}_j+\mathbf{q}_i^T\boldsymbol{\mu}_k+\boldsymbol{\mu}_q^T\boldsymbol{\mu}_k, \tag{8}$$

where $\boldsymbol{\mu}_q$ and $\boldsymbol{\mu}_k$ denote $\frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{q}_i$ and $\frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{k}_j$, respectively.

Note that the last two terms ($\mathbf{q}_i^T\boldsymbol{\mu}_k$ and $\boldsymbol{\mu}_q^T\boldsymbol{\mu}_k$) are factors in common with both the numerator and denominator of the correlation function f and the normalization factor $\mathcal{C}$, so they can be eliminated as follows:

$$\begin{aligned}
&\frac{\exp\left(\mathbf{q}_i^T\mathbf{k}_j\right)}{\sum_{t=1}^{N_p}\exp\left(\mathbf{q}_i^T\mathbf{k}_t\right)}\\
&=\frac{\exp\left(\left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_j-\boldsymbol{\mu}_k\right)+\boldsymbol{\mu}_q^T\mathbf{k}_j+\mathbf{q}_i^T\boldsymbol{\mu}_k+\boldsymbol{\mu}_q^T\boldsymbol{\mu}_k\right)}{\sum_{t=1}^{N_p}\exp\left(\left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_t-\boldsymbol{\mu}_k\right)+\boldsymbol{\mu}_q^T\mathbf{k}_t+\mathbf{q}_i^T\boldsymbol{\mu}_k+\boldsymbol{\mu}_q^T\boldsymbol{\mu}_k\right)}\\
&=\frac{\exp\left(\left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_j-\boldsymbol{\mu}_k\right)+\boldsymbol{\mu}_q^T\mathbf{k}_j\right)\exp\left(\mathbf{q}_i^T\boldsymbol{\mu}_k+\boldsymbol{\mu}_q^T\boldsymbol{\mu}_k\right)}{\sum_{t=1}^{N_p}\exp\left(\left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_t-\boldsymbol{\mu}_k\right)+\boldsymbol{\mu}_q^T\mathbf{k}_t\right)\exp\left(\mathbf{q}_i^T\boldsymbol{\mu}_k+\boldsymbol{\mu}_q^T\boldsymbol{\mu}_k\right)}\\
&=\frac{\exp\left(\left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_j-\boldsymbol{\mu}_k\right)+\boldsymbol{\mu}_q^T\mathbf{k}_j\right)}{\sum_{t=1}^{N_p}\exp\left(\left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_t-\boldsymbol{\mu}_k\right)+\boldsymbol{\mu}_q^T\mathbf{k}_t\right)}.
\end{aligned} \tag{9}$$

Finally, we obtain

$$\sigma(\mathbf{q}_i^T\mathbf{k}_j)=\sigma(\underbrace{\left(\mathbf{q}_i-\boldsymbol{\mu}_q\right)^T\left(\mathbf{k}_j-\boldsymbol{\mu}_k\right)}_{pairwise}+\underbrace{\boldsymbol{\mu}_q^T\mathbf{k}_j}_{unary}). \tag{10}$$

4 Experiment settings

Semantic Segmentation. We mostly follow [6] in training and inference.
Trainng. The SGD optimizer with poly learning rate policy $(1-(\frac{iter}{iter_{max}})^{0.9})$ is employed. For Cityscapes, the networks are trained on 4 GPUs with 2 images per GPU for 60K iterations. The initial learning rate is 0.01, the weight decay is 0.0005. Input images are cropped to 769×769. For ADE20K, the networks

are trained on 8 GPUs with 2 images per GPU for 150K iterations. The initial learning rate is 0.02, and the weight decay is 0.0001. Input images are cropped to 520×520. For PASCAL-Context, the network is trained on 4 GPUs with 4 images per GPU for 30K iterations. We initialize backbone models by the ImageNet-pretrained weights, and randomly initialize new layers in the segmentation head, including the attention module and the classification layer. The initial learning rate is 0.001, and the weight decay is 0.0001. Input images are cropped to 520×520. For all datasets, the data is augmented with random horizontal flipping, random scaling within $[0.5, 2.0]$, and random brightness jittering of $[-10, 10]$. Following [9], online hard example mining (OHEM) and an auxiliary loss on the output of conv4 with a weight of 0.5 are employed for Cityscapes and ADE20K, only auxiliary loss is employed for PASCAL-Context.

Inference. For Cityscapes, we sample 769×769 windows for inference and their results are fused to generate the prediction of an entire image. For other datasets, we resize the image resolution to be the same as in training and a multi-scale test is adopted.

Object Detection. We use the standard configuration of Mask R-CNN [5] with FPN and ResNet-50 as the backbone architecture, and report the mean average-precision scores at different boxes and the mask IoUs on the COCO2017 validation set. The input images are resized such that their shorter side is 800 pixels [7]. We trained on 4 GPUs with 4 images per GPU (effective mini batch size of 16). The backbones of all models are pretrained on ImageNet classification [3], then all layers except for stage1 and stage2 are jointly fine-tuned.

In training, synchronized BatchNorm is adopted, and the learning rate scheduler follows the $1\times$ settings of 12 epochs in [5] where the initial learning rate is 0.02 and decayed by a factor of 10 at the 8^{th} and 11^{th} epoch. The weight decay is 0.0001 and momentum is 0.9.

Action Recognition. We adopt the slow-only baseline in [4], the best single model to date that can utilize weights inflated [2] from the ImageNet pretrained model. All the experiment settings follow the slow-only baseline in [4], where 8 frames (8×8) are used as input, and 30-clip validation is adopted. Following [8], we insert (disentangled) non-local blocks after every two residual blocks.

5 Statistic results on COCO dataset

In this section, we measure the averaged consistency measures of the attention maps to ground-truth region maps on COCO dataset. On COCO object detection dataset, the pairwise and unary terms alone are also learnt to have clearly separate visual meanings: the former one shows some within-region meaning while the latter one mostly focuses on salient area, but not limited to boundaries. Quantitatively, we have statistical results in Table 2. The results indicate NL module performs no better than random baseline in learning these two clues, while DNL learns two clues to some extent.

Another issue of NL on COCO is that the pairwise term is almost hindered by the unary, which is also observed by [1]). Following [1], we we count the

Table 2. Statistics results between attention maps of the non-local variants and the ground-truth within-category and boundary maps on COCO dataset

method	pair ∩ within-category	pair ∩ boundary	unary ∩ boundary
random	0.231	0.140	0.153
NL (Eq. 2)	0.297	0.135	0.170
DNL (Eq. 12)	0.483	0.141	0.323

Table 3. Average cosine distance between attention maps of queries on COCO dataset.

method	input	output	attention
NL (Eq. 2)	0.397	0.0008	0.0017
DNL (Eq. 12)	0.403	0.163	0.245

average cosine distance between attention maps of queries. It is close to 0 (the right column) by NL, indicating the degeneration to a unary term. DNL well addresses this issue as show in Table 3

6 More Examples of Learnt Attention Maps by NL/DNL Methods

In this section, we show more examples of the learnt attention maps by the NL/DNL methods on the Cityscapes semantic segmentation, COCO object detection/instance segmentation and Kinetics action recognition tasks.

Fig. 1 show more examples of the learnt attention maps by NL/DNL on Cityscapes. With DNL block, the whitened pairwise term learns clear within-region clues while the unary term learns salient boundaries, which cannot be observed in that of the original NL block.

Fig. 2 show more examples of the learnt attention maps by NL/DNL on COCO object detection/instance segmentation. It can be seen that the attention maps of NL block are mainly dominated by the unary term that different query points (marked in red) have similar overall attention maps. In DNL, the pariwise term in DNL shows clear within-region meaning and appears significant in the final overall attention maps, that different query points have different overall attention maps. DNL also shows more focus to salient regions than the one in an NL block.

Fig. 3 show more examples of the learnt attention maps by NL/DNL on Kinetics action recognition task. 4 frames in a video clip are visualized. The unary term of DNL shows better focus to salient regions than the one in an NL block. The pairwise term in DNL also shows clearer within-region meaning than that in an NL block.

References

1. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. arXiv preprint arXiv:1904.11492 (2019)

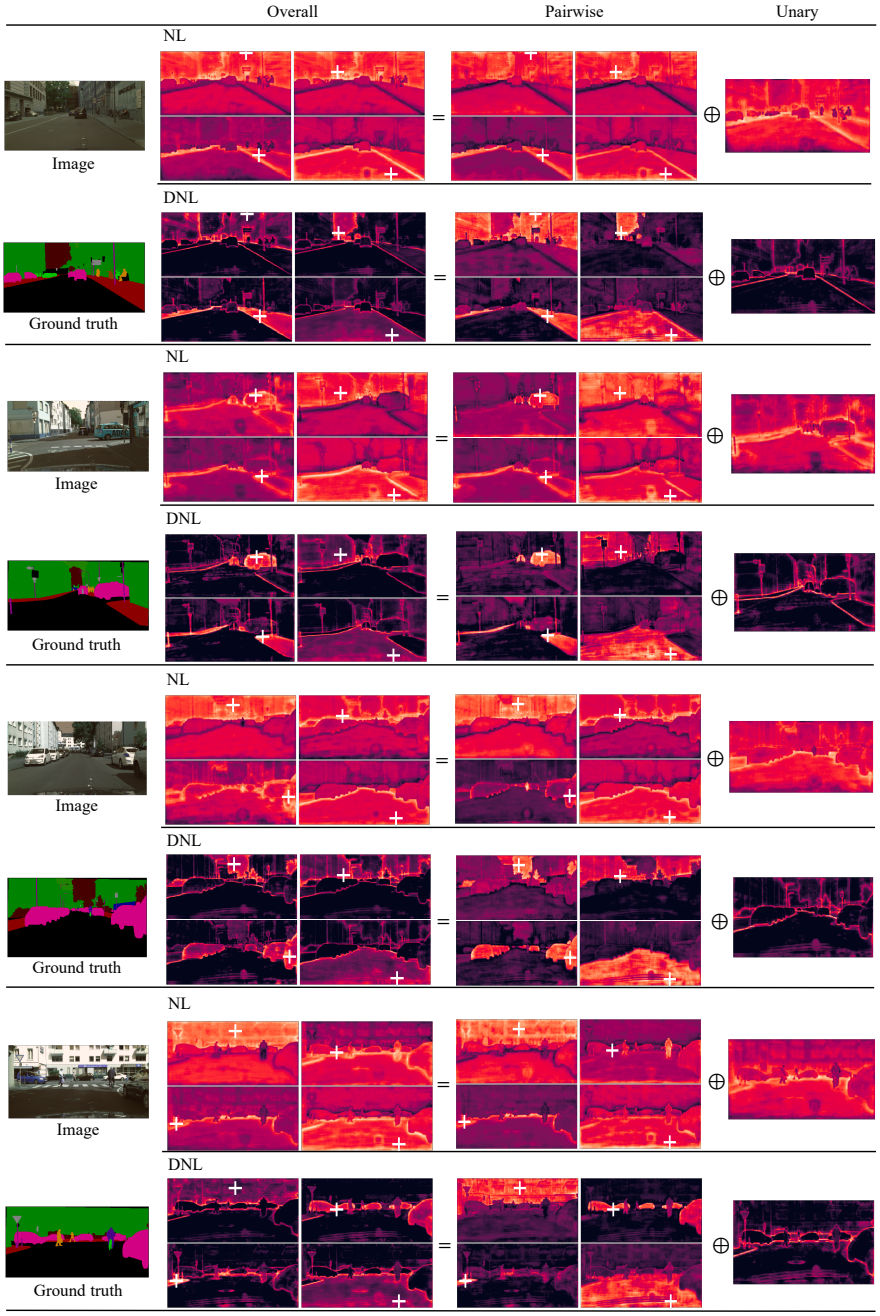

Fig. 1. Visualization of attention maps of NL block and our DNL block on Cityscapes Dataset. The query points are marked in white cross

Fig. 2. Visualization of attention maps of NL block and our DNL block on COCO object detection task. The query points are marked in red.

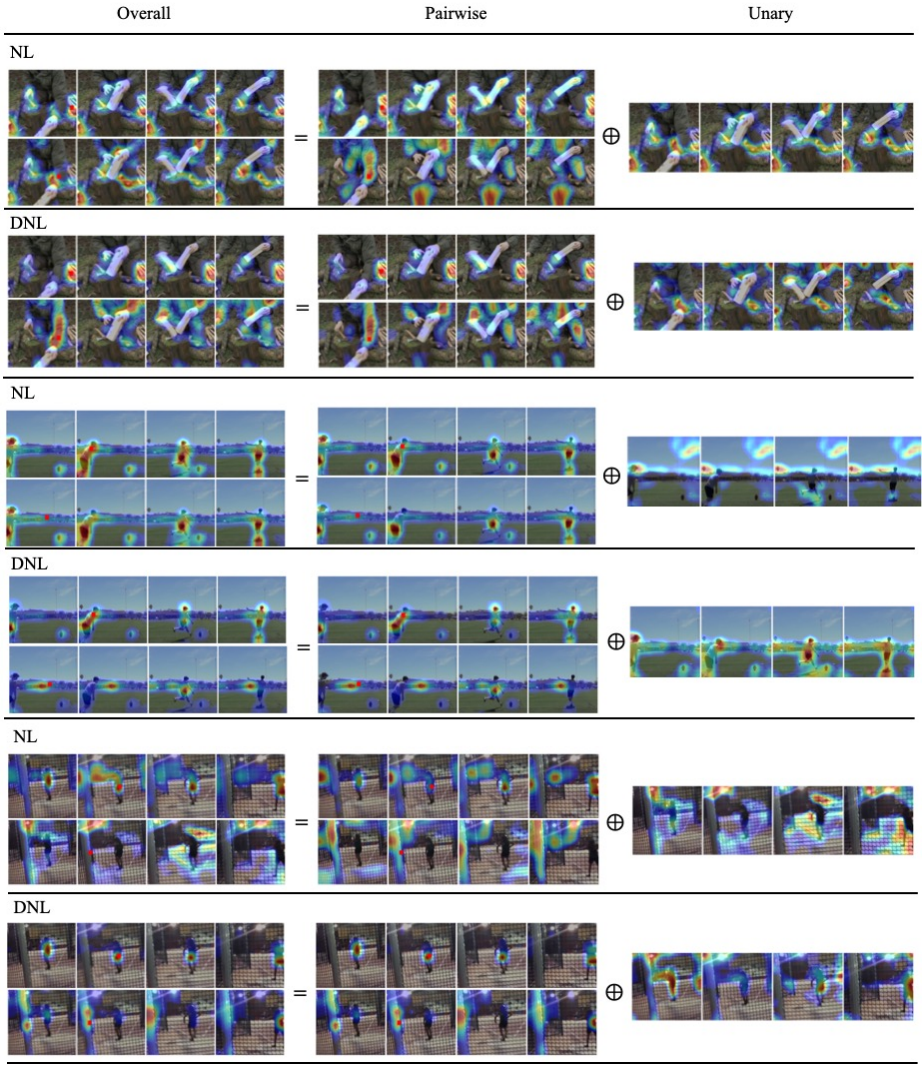

Fig. 3. Visualization of attention maps of NL block and our DNL block on Kinetics action recognition. 4 frames of a video clip are visualized. For each sample of each block, two different queries are chosen as the top and bottom rows. The query points are marked in red

2. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
4. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. arXiv preprint arXiv:1812.03982 (2018)
5. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
6. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 603–612 (2019)
7. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
8. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7794–7803 (2018)
9. Yuan, Y., Wang, J.: Ocnet: Object context network for scene parsing. arXiv preprint arXiv:1809.00916 (2018)