Supplementary Material for "Combined Depth Space Based Architecture Search For Person Re-identification"

Hanjun Li[1,4], Gaojie Wu[1], Wei-Shi Zheng[1,2,3,*]
[1]School of Computer Science and Engineering, Sun Yat-sen University, China
[2]Peng Cheng Laboratory, Shenzhen 518005, China
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
[4]Pazhou Lab, Guangzhou, China
lihj85@mail2.sysu.edu.cn, wugj7@mail2.sysu.edu.cn,wszheng@ieee.org

1. Details for Top-k Sample Search

Different from normal NAS, we aim to search for most suitable network architectures for ReID. Therefore, we directly search the network architectures on Market1501 [10]. In Market1501, the dataset is divided into a training set with 12936 images of 751 persons and a testing set of 750 persons containing 3368 query images and 19732 gallery images. According to the proposed Top-k Sample Search, we need to split the training set into a new training set and a validation set. For each identity, we select 4 images to construct the validation set. We utilize the triplet sampler as in [5] on both training set and validation set to prepare the batch data. Particularly, we choose the batch size as 64 and the number of identities per batch is 16, namely each identity has 4 instances per batch. SGD optimizer with weight decay 3e-4 and Adam optimizer with weight decay 0.001 are utilized for the network parameters and the architecture parameters optimization respectively. The network parameters learning rate η_w starts from 0.025 and decays to 0.0001 via a cosine lr_scheduler while architecture parameters learning rate η_α starts from 3e-4 and decayed by 0.1 at 80, 160 epochs. Totally, we train the search network for 240 epochs. During the training process, softmax loss and triplet loss are jointly utilized for optimization and the margin of triplet loss is 0.3.

In Tab. 1, we show the inner structures of CNet and CDNet via Top-k Sample Search, where $k \in \{1, 2, 3, 4\}$. As analyzed in main manuscript (section 4.1), the architectures searched via top-2 sample search almost achieve the best performance. As for CNet, the best architecture (top-2 CNet) tends to select large kernel combination thus makes up the defect of shallow depth. Note that the depth of CDNet can vary from 6 to 12 and none of the searched architectures choose the biggest depth. As we can see, the best architecture of CDNet(top-2 CDNet) just add one more CBlock at each stage. Since top-2 CDNet has enough depth, its kernels size tend to be smaller compared with top-2 CNet. Therefore, it is inappropriate to randomly add the depth of the network.

*Corresponding author

2. Visualization for the effect of BLNeck

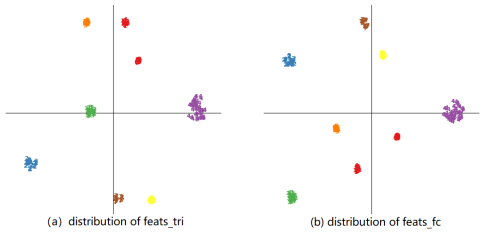

Figure 1. We first select 8 identities and use t-sne method to project their features into 2-dimension space. The feats_tri constrained by triplet loss are on the left side and the feats_fc constrained by softmax loss are on the right side.

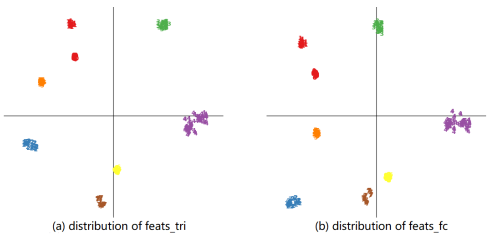

Figure 2. The feats_tri constrained by triplet loss are on the left side and the feats_fc constrained by softmax loss are on the right side.

To further understand the influence of BLNeck, we visualize the distribution of the features which are processed

Layer	CNet(k_1, k_2)				CDNet(k_1, k_2, r)			
	top-1	top-2	top-3	top-4	top-1	top-2	top-3	top-4
1	(3,5)	(5,7)	(5,7)	(5,7)	(3,7,2)	(3,5,1)	(5,7,2)	(3,9,2)
2	(7,9)	(7,9)	(3,9)	(3,5)	(3,9,1)	(3,7,2)	(5,7,2)	(5,9,1)
3	(3,7)	(7,9)	(3,5)	(5,7)	(5,9,2)	(5,7,2)	(3,5,2)	(5,7,2)
4	(3,9)	(7,9)	(3,5)	(3,7)	(3,7,1)	(5,9,1)	(5,7,1)	(3,7,1)
5	(5,9)	(7,9)	(3,5)	(3,9)	(3,9,2)	(5,7,2)	(7,9,2)	(5,7,1)
6	(7,9)	(3,5)	(5,7)	(3,7)	(7,9,2)	(5,7,1)	(3,7,1)	(5,7,1)
Depth	6	6	6	6	10	9	10	8

Table 1. The inner structures of all searched architectures. k_1, k_2 denote the kernel size of two branches in CBlock respectively. r denotes the repeated times of CBlock for CDBlock

before and after by BLNeck. As shown in Fig. 1, BLNeck learns to map the feats_tri to another embedding space which is fitter to softmax loss. According to the distribution of feats_fc, there are clear angle margins between each identity. BNNeck proposed in [5] can easily balance the constraint of triplet loss and triplet loss. In Fig. 2, we show that BNNeck only makes a little adjustment of the features, and they still affect greatly by each other, thus the training can not converge peacefully. Compared with BNNeck, the proposed BLNeck has stronger ability to learn the mapping from embedding space constrained by triplet loss to that constrained by softmax loss.

Architecture	Param(M)	rank-1	mAP
+Softmax	2.2	92.5	80.0
+Softmax_Triplet	2.2	93.1	82.2
+FBLNeck	1.7	**93.6**	**83.4**

Table 2. Effect of FBLNeck. The backbone is the body of OSNet. +Softmax denotes OSNet is trained with only softmax loss. +Softmax_Triplet denotes that OSNet is trained with softmax and triplet loss. +FBLNeck denotes that OSNet is trained with FBLNeck as its attached head. All experiments are conducted by us on Market1501.

3. Scaling CDNet with width multiplier and resolution multiplier

As shown in Tab. 3, we scale the CDNet for specific devices with limited computational resources via adjusting the width multiplier β and resolution multiplier γ. When fixing the β and shrinking the γ from 1.0 to 0.5, the number of FLOPs decreases significantly while the rank-1 just drops smoothly. However, we find that both rank-1 and mAP drop dramatically when β decreases to 0.25. This is because the resolution of images are reduced to 4×2 at stage 3, which is too small to learn effective information via convolution at stage 3. It is worth noting that CDNet still can achieve 91.7%/79.7% at rank-1/mAP when both β and γ are set to 0.25 and 1.0 respectively (with merely 0.1M parameters and 77.9M FLOPs). This suggests that CDNet has great potential for deployment in edge devices such as surveillance cameras with limited computational resources. On the right side, we also report the result of OSNet accordingly. Apparently, CDNet outperforms OSNet at rank-1 and mAP for most settings with lower parameters and FLOPs, which demonstrates the robustness of combined pattern learning.

4. Implementation of FBLNeck in OSNet

As analyzed in main manuscript (section 4.5), FBLNeck could take advantage of the combination of triplet loss and softmax loss and utilize fine-grained information, which leads to high performance. As a newly proposed neck, FBLNeck can be inserted to other models easily in addition to CNet and CDNet. Here we investigate whether inserting FBLNeck to OSNet could enhance the performance of OSNet [11]. As shown in Tab. 2, with FBLNeck implemented with OSNet, both rank-1 accuracy and mAP increase, especially for mAP. It is fair to say that the proposed FBLNeck could better balance the effect of triplet loss and softmax loss to help models achieve higher performance. Note that the number of parameters of OSNet+FBLNeck is reduced to 1.7M, since we remove the attached head of OSNet and our FBLNeck is removable at inference time. Moreover, comparing OSNet + FBLNeck and CDNet, our searched CDNet outperforms OSNet with less parameters, which means that our proposed search algorithm can obtain models that are computationally efficient and suitable for ReID.

5. Evaluation on ImageNet

In this section, we evaluate the transferability of proposed CDNet on the ImageNet [6]. The size of image is resized to 224×224. Random horizontal flip and random crop are utilized for data augmentation. We adopt the training scheme as in [4]. CDNet is trained for 240 epochs with weight decay 3e-5 and initial SGD learning rate 0.1(decayed by a factor of 0.97 after each epoch). As shown in Tab. 4, although our CDNet is originally designed for ReID, it still achieves comparable performance among lightweight

β	γ	CDNet				OSNet			
		Param(M)	FLOPS(M)	rank-1	mAP	Param(M)	FLOPs(M)	rank-1	mAP
1.0	1.0	1.8	948.8	**95.1**	**86.0**	2.2	978.9	94.8	84.9
1.0	0.75	1.8	533.7	**94.7**	**85.0**	2.2	550.7	94.4	83.7
1.0	0.5	1.8	237.3	**93.3**	**82.3**	2.2	244.9	92.0	80.3
1.0	0.25	1.8	59.4	**86.9**	**69.5**	2.2	61.5	**86.9**	67.3
0.75	1.0	1.0	552.3	**94.7**	**85.1**	1.3	571.8	94.5	84.1
0.75	0.75	1.0	310.7	94.2	**84.3**	1.3	321.7	**94.3**	82.4
0.75	0.5	1.0	138.1	**93.1**	**81.4**	1.3	143.1	92.9	79.5
0.75	0.25	1.0	34.6	**86.8**	**69.3**	1.3	35.9	85.4	65.5
0.5	1.0	0.5	262.0	**93.4**	**83.8**	0.6	272.9	**93.4**	82.6
0.5	0.75	0.5	147.4	**93.9**	**83.5**	0.6	153.6	92.9	80.8
0.5	0.5	0.5	65.5	**92.5**	**80.3**	0.6	68.3	91.7	78.5
0.5	0.25	0.5	16.4	84.3	**66.4**	0.6	17.2	**85.4**	66.0
0.25	1.0	0.1	77.9	91.7	**79.7**	0.2	82.3	**92.2**	77.8
0.25	0.75	0.1	43.8	**91.8**	**78.8**	0.2	46.3	91.6	76.1
0.25	0.5	0.1	19.5	**89.3**	**75.0**	0.2	20.6	88.7	71.8
0.25	0.25	0.1	4.88	**79.6**	**59.4**	0.2	5.2	79.1	56.0

Table 3. Results(%) of varying width multiplier β and resolution multiplier γ for CDNet and OSNet. The resolution is 256×128, 192×96, 128×64 and 64×32 for $\gamma = 1.0, \gamma = 0.75, \gamma = 0.5$ and $\gamma = 0.25$ respectively.

Model	Param(M)	FLOPs(M)	Top-1
CARS[9]	5.1	519	75.2
FBNet[8]	5.5	375	74.9
GDAS[1]	5.3	581	74.0
DARTS[4]	4.7	574	73.3
OSNet[11]	2.7	1511	**75.5**
GhostNet[2]	5.2	141	73.9
MobileNetV2[7]	3.4	300	73.0
MobileNetV3[3]	5.4	219	75.2
CDNet(ours)	**2.5**	1571	75.1

Table 4. Top-1(%) accuracy on ImageNet-2012 validation set.

networks which are specially designed for classification. In particular, CDNet outperforms MobileNetV2 by 2.1% with fewer number of parameters. It is worth noting that CDNet surpasses GDAS and DARTS by 1.1% and 1.8% respectively, which indicates that the proposed CDS also has great potential for classification tasks. Obviously, the superior performance on classification tasks demonstrates the benefit of learning combined pattern information.

6. Inference time on Market1501

As shown in Tab. 5, with about $3\times$ fewer FLOPs and $13\times$ fewer parameters, CDNet achieves competitive performance with lower latency compared with BagofTrick, which is representative of those models utilizing ResNet as backbone. Besides, compared with the other two lightweight models, CDNet achieves the best performance with faster speed.

Model	Param(M)	Flops(M)	Times(s)	Rank-1	mAP
BagofTrick [5]	25.1	4053.3	248.3	94.5	85.9
OSNet[11]	2.2	979.0	156.0	94.8	84.9
GDAS[1]	4.0	1109.2	150.9	89.1	73.2
CDNet(ours)	1.8	955.1	142.4	95.1	86.0

Table 5. All experiments are conducted on Market1501 with single RTX 2080 and the batch size is 128. The time is the average of 5 times inference time on test set with 19281 images.

References

[1] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 1761–1770, 2019. 3

[2] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations, 2020. 3

[3] Andrew Howard, Ruoming Pang, Hartwig Adam, Quoc Le, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, and Yukun Zhu. Searching for mobilenetv3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019. 3

[4] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018. 2, 3

[5] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification, 2019. 1, 2, 3

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge.

International Journal of Computer Vision, 115(3):211–252, 2015. 2

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3

[8] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019. 3

[9] Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. Cars: Continuous evolution for efficient neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1829–1838, 2020. 3

[10] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015. 1

[11] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3702–3712, 2019. 2, 3