Consistent-Aware Deep Learning for Person Re-identification in A Camera Network Supplementary Materials

Anonymous CVPR submission

Paper ID 2458

In the supplementary materials, we first provide the full equations for the general cases where different persons appear in different cameras. Secondly we show the detailed structure of the pretrained CNN used in our paper. Thirdly we provide a comparison between different pretrained models applied to person re-identification and explain the reasons for choosing models. Lastly, we provide some of the detailed results which are too long to be put on our paper.

1. The Full Equations of General Cases

In section 3.3 of our paper, we extended our framework to general cases where different persons appear in different cameras. We showed that this can be done by modifying the row and column constraints. Here we provide the full objective and gradient for general cases. The modified objective is:

$$\begin{aligned}\min_{\mathbf{H}} \mathbf{J_1'} &= \sum_{a,b=1}^{m} (-\|\mathbf{H}^{a,b} \cdot \mathbf{C}^{a,b}\|_F^2) \\ &+ \sum_{a,b=1}^{m} (\alpha \mathbf{J_B}^{a,b} + \beta(\mathbf{J_R'}^{a,b} + \mathbf{J_C'}^{a,b})) \\ &+ \mu \frac{1}{m-2} \sum_{a,c,b}^{m} \mathbf{J_T}^{a,c,b}\end{aligned} \tag{1}$$

where

$$\begin{aligned}&\mathbf{J_B} = \|(\mathbf{H}-0.5)\cdot(\mathbf{H}-0.5) - 0.25\|_F^2 \\ &\mathbf{J_R}' = (\|(\mathbf{H}e-0.5e)\cdot(\mathbf{H}e-0.5e) - 0.25e\|_2^2 \\ &\mathbf{J_C}' = \|(e^T\mathbf{H}-0.5e^T)\cdot(e^T\mathbf{H}-0.5e^T) - 0.25e^T\|_2^2 \\ &\mathbf{J_T}^{a,c,b} = \|\max\{0, \mathbf{H}^{a,c}\mathbf{H}^{c,b} - \mathbf{H}^{a,b}\}\|_F^2\end{aligned} \tag{2}$$

The terms $\mathbf{J_R'}$ and $\mathbf{J_C'}$ are modified, as in general cases the sum of each row and column might be 0 or 1. Accordingly, we provide the corresponding gradient of the equation above:

$$\begin{aligned}\frac{\partial \mathbf{J_1'}}{\partial \mathbf{H}^{a,b}} &= -\mathbf{H}^{a,b} \cdot \mathbf{C}^{(a,b)2} \\ &+\alpha((\mathbf{H}^{a,b}-0.5)^{\cdot 2} - 0.25)\cdot(\mathbf{H}^{a,b}-0.5) \\ &+\beta((\mathbf{H}^{a,b}e-0.5e)\cdot(\mathbf{H}^{a,b}e-0.5e) - 0.25e) \\ &\quad \cdot(\mathbf{H}^{a,b}e-0.5e)e^T \\ &+\beta e((e^T\mathbf{H}^{a,b}-0.5e^T)\cdot(e^T\mathbf{H}^{a,b}-0.5e^T) - 0.25e^T) \\ &\quad \cdot(e^T\mathbf{H}^{a,b}-0.5e^T) \\ &+\mu\sum_{c}^{m} -(max\{0, \mathbf{H}^{a,c}\mathbf{H}^{c,b} - \mathbf{H}^{a,b}\})\end{aligned} \tag{3}$$

We can use the modified gradient to update $\mathbf{H}$ to find the globally optimal matching.

Notice that, in some scenarios, we conduct experiments under query-gallery settings. In this case, the row and column indexes of matrices $\mathbf{H}$ and $\mathbf{C}$ refer to query and gallery respectively. We assume that row indexes refer to query while column indexes refer to gallery. To make it clear, we can denote $\mathbf{H}$ as $\mathbf{H}^{a_q,b_g}$, where the subscript q refers to query images while g refers to gallery images. In this case, we cannot simply compute $\mathbf{J^T}$ as in equation 2, as it will lead to mismatching between query and gallery. Instead, we modified the term $\mathbf{H}^{a_q,c_g}\mathbf{H}^{c_q,b_g}$ as:

$$\mathbf{H}^{a_q,c_g}\mathbf{H}^{c_g,c_q}\mathbf{H}^{c_q,b_g} \tag{4}$$

We add a term $\mathbf{H}^{c_g,c_q}$ between the two $\mathbf{H}$s, and the subscripts conform to each other. The modified $\mathbf{J_T'}$ can be formulated as:

$$\mathbf{J_T'}^{a,c,b} = \|\max\{0, \mathbf{H}^{a,c}\mathbf{H}^{c,c\mathbf{T}}\mathbf{H}^{c,b} - \mathbf{H}^{a,b}\}\|_F^2 \tag{5}$$

We can extend our framework to query-gallery settings (like the Market-1501) by replace $\mathbf{J_T}$ with $\mathbf{J_T'}$ in 1.

2. The Detailed Structure of the CNN Model

In our paper, we performed our method by fine-tuning pretrained model using CADL. We chose the pretrained

Table 1. The detailed structure of the pretrained CNN model.

Name	patch size/ stride	output size	#1×1	#3×3 reduce	#3×3	double #3×3 reduce	double #3×3	pool+proj
input		3×144×56						
conv1 - conv3	3×3/2	32×144×56						
pool3	2×2/2	32×72×28						
inception (4a)		256×72×28	32	32	32	32	32	avg + 32
inception (4b)	stride 2	384×72×28	32	32	32	32	32	max + pass through
inception (5a)		512×36×14	64	64	64	64	64	avg + 64
inception (5b)	stride 2	768×36×14	64	64	64	64	64	max + pass through
inception (6a)		1024×36×14	128	128	128	128	128	avg + 128
inception (6b)	stride 2	1536×36×14	128	128	128	128	128	max + pass through
fc7		256						

model provided in [1]. The CNN takes input of size 144×56. It starts with 4 concatenated convolution layers followed by a pooling layer. Next is a series of 6 inception units. After the final fully connected layer, the CNN produces features of length 256. The detailed structure of the pretrained CNN is listed in Table 1 (taken from [1]).

3. Comparison of Different CNN Structures

Unlike the model above, most of the models pretrained on ImageNet [2], like VGG [3], GoogLeNet [4], etc., take input of square size. However, due to the natural characteristics of human body, the shape of pedestrian image is a rectangle with aspect ratio smaller than 0.5. In this case, it would be improper to resize the person image into a square, which changes the patterns of original images significantly and can lead to further computation cost. Cropping a square region out of the image is even worse, as it will lose more than half of the information. If we feed these resized or cropped images into neural networks, it may lead to poor performance and useless computation cost.

To prove our idea, we compared the pretrained model we used (input size 144 × 56) to the pretrained GoogLeNet model on ImageNet (input size 224 × 224) from caffe [5] model zoo. They share a lot of similarities in network structure while differ a lot in input size. We fine-tuned them on another popular person re-identification dataset *CUHK03* [6] using contrastive loss and followed all the same settings. We followed the protocol in [7], where 100 persons were randomly selected for testing, 100 for validation and the rest 1160 for training. We set batch size to 30 pairs and train for 30,000 batches. The initial learning rate was set to 0.01 and reduced by a factor of 0.1 after each 10,000 batches. The margin for the contrastive loss was kept as 1.0.

The results are shown in Table 2. Even the GoogLeNet is actually deeper than the selected model, results show that selected model from [1] outperforms GoogLeNet on CUHK03. This shows the influence of input size in re-identification problem. So we finally chose the model from [1] as our pretrained model.

Table 2. Comparison of the 2 models.

Model	Rank 1	Rank 5	Rank 10	final loss
GoogLeNet	40.8%	76.1%	89.1%	0.027
Ours	63.6%	92.0%	95.9%	0.0078

4. Detailed Experimental Results on Market-1501

Due to the limited length, we only provided the average accuracy and variance of 30 camera pairs in the paper. Here, we offer the full results of the Market-1501 dataset for our protocol.

In our protocol, we measured the accuracy for all 30 camera pairs. We offer the full results for **Ours - Pretrained**, **Ours - Contrastive**, **Ours - Cosine** and **Ours - CADL** with both single-query and multi-query settings in Table 3, 4, 5, 6, 7, 8, 9, 10 respectively. All the experiments were conducted by using gallery images to match query images, so the matching accuracy of **Cam1 - Cam2** is different to **Cam2 - Cam1**. In the tables, the row camera IDs refer to query images, and the column camera IDs refer to gallery images.

References

[1] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. *arXiv preprint arXiv:1604.07528*, 2016. 2

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2

Table 3. The full results of **Ours - Pretrained (SQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	28.59	25.15	42.22	19.01	23.80
Cam2	26.89	-	31.25	25.00	19.51	26.33
Cam3	22.80	33.15	-	32.59	41.26	23.22
Cam4	35.37	30.89	26.83	-	33.33	31.71
Cam5	15.00	23.06	36.29	34.84	-	18.55
Cam6	15.08	17.50	13.00	35.18	17.33	-

Table 4. The full results of **Ours - Contrastive (SQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	40.72	52.25	26.20	47.60	42.66
Cam2	51.52	-	56.25	20.27	47.54	46.21
Cam3	52.73	46.29	-	21.54	53.29	48.67
Cam4	52.03	39.84	49.19	-	58.13	36.59
Cam5	50.16	39.03	54.68	25.48	-	43.39
Cam6	47.31	40.21	52.86	20.28	45.93	-

Table 5. The full results of **Ours - Cosine (SQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	55.54	50.45	71.56	44.31	48.35
Cam2	53.98	-	60.04	62.50	51.14	53.79
Cam3	51.47	60.70	-	64.62	61.96	52.73
Cam4	56.50	50.41	48.78	-	50.41	53.25
Cam5	41.45	53.39	58.71	66.77	-	46.94
Cam6	46.97	50.43	49.74	70.71	44.71	-

Table 6. The full results of **Ours - CADL (SQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	65.27	51.35	87.57	50.60	56.14
Cam2	57.77	-	60.23	85.23	57.39	61.17
Cam3	57.90	72.03	-	84.48	69.23	61.96
Cam4	57.32	66.26	50.81	-	52.03	67.48
Cam5	40.97	65.16	58.39	82.10	-	51.13
Cam6	51.13	65.51	49.05	85.27	47.14	-

Table 7. The full results of **Ours - Pretrained (MQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	42.51	51.35	39.37	32.19	30.69
Cam2	43.18	-	55.11	29.55	28.41	30.68
Cam3	47.69	50.77	-	29.51	64.90	35.80
Cam4	58.54	36.59	41.87	-	50.00	36.99
Cam5	30.97	29.35	65.16	40.32	-	26.61
Cam6	26.86	25.48	28.08	44.54	24.09	-

Table 8. The full results of **Ours - Contrastive (MQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	51.65	70.36	28.74	63.17	58.83
Cam2	60.80	-	74.43	24.24	56.63	57.20
Cam3	65.73	55.38	-	25.31	67.55	61.82
Cam4	67.89	43.50	60.98	-	66.26	47.97
Cam5	65.97	48.87	76.61	30.81	-	56.94
Cam6	68.11	51.82	74.35	23.74	63.43	-

Table 9. The full results of **Ours - Cosine (MQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	67.81	76.35	72.01	64.97	70.21
Cam2	71.59	-	85.23	65.72	68.94	70.83
Cam3	74.97	74.83	-	66.85	81.54	73.57
Cam4	77.64	61.38	71.54	-	70.73	63.01
Cam5	68.39	68.39	86.77	70.32	-	67.90
Cam6	69.84	68.28	75.74	67.24	66.38	-

Table 10. The full results of **Ours - CADL (MQ)**.

	Cam1	**Cam2**	**Cam3**	**Cam4**	**Cam5**	**Cam6**
Cam1	-	81.74	77.54	93.41	71.26	80.99
Cam2	77.27	-	86.55	92.05	80.30	81.82
Cam3	80.00	89.93	-	92.17	90.21	83.36
Cam4	79.67	79.27	75.20	-	72.76	80.08
Cam5	70.65	83.39	88.06	90.32	-	76.13
Cam6	75.74	83.88	76.78	92.03	74.00	-

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015. 2

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 2

[6] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, pages 152–159, 2014. 2

[7] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, pages 791–808, 2016. 2