# Appendix

## 1 Role of Random Occlusion

The random occlusion (RO) we designed for data augmentation is similar to the random erasing (RE) [6] and cutout [2] methods. In the RE implementation, the target erasing area is sampled from a combination of random area and aspect ratio, which could exceed the original image height or width. Therefore, it needs to try multiple times (100 by default) to generate a reasonable region for erasing. In contrast, in our implementation of the random occlusion, a square area is used, with the size randomly sampled at most $0.8 \times width$ of the image, and randomly put in a valid location. Then the square area is filled with white pixels. Note that with a simple square area, there is no need to sample multiple times of areas and aspect ratios and check the validity, and hence the generation process is more efficient. As for the cutout method, it uses multiple square regions in fixed sizes specified by hyperparameters, but not in random. The fixed-size regions may make the cut either too small or too large, and so it is not very convenient to set.

To show their differences, in the training of QAConv, we compare these data augmentation methods as well as a baseline without any random occlusion. From the results shown in Table 1, it can be observed that the three data augmentation methods generally improve the baseline which does not apply any random occlusion. Intuitively, they are useful for QAConv because random occlusion forces QAConv to learn various local correspondences, instead of only salient but easy ones. Besides, the three data augmentation methods perform comparable, with the RO implementation being slightly better. Therefore, considering also the efficiency of the RO implementation, it is adopted in the training of the proposed QAConv algorithm.

**Table 1.** Role of random occlusion.

| Method | Market→Duke | | Duke→Market | |
|---|---|---|---|---|
| | Rank-1 | mAP | Rank-1 | mAP |
| QAConv without occlusion | 50.5 | 29.5 | 61.6 | 28.4 |
| QAConv with RE [6] | 51.6 | 30.6 | 62.0 | 29.8 |
| QAConv with cutout [2] | 51.6 | 30.8 | 62.6 | 30.3 |
| QAConv with RO | **54.4** | **33.6** | **62.8** | **31.6** |

## 2 Complete Comparisons of Backbone Networks

Tables 2 and 3 show complete comparisons between the QAConv results with the ResNet-50 as backbone (denoted as $\text{QAConv}_{50}$) and with the ResNet-152

as backbone (denoted as $\text{QAConv}_{152}$), with DukeMTMC-reID and Market-1501 as the target datasets, respectively. Results of applying re-ranking alone are not shown in the main paper.

**Table 2.** Comparison (%) of backbone networks with DukeMTMC-reID as the target dataset.

| Method | Training | | Test: Duke | |
|---|---|---|---|---|
| | Source | Target | R1 | mAP |
| $\text{QAConv}_{50}$ | Market | | 48.8 | 28.7 |
| $\text{QAConv}_{152}$ | Market | | 54.4 | 33.6 |
| $\text{QAConv}_{50}$ + RR | Market | | 56.9 | 47.8 |
| $\text{QAConv}_{152}$ + RR | Market | | 61.8 | 52.4 |
| $\text{QAConv}_{50}$ + RR + TLift | Market | | 64.5 | 55.1 |
| $\text{QAConv}_{152}$ + RR + TLift | Market | | 70.0 | 61.2 |
| $\text{QAConv}_{50}$ | MSMT | | 69.4 | 52.6 |
| $\text{QAConv}_{152}$ | MSMT | | 72.2 | 53.4 |
| $\text{QAConv}_{50}$ + RR | MSMT | | 76.7 | 71.2 |
| $\text{QAConv}_{152}$ + RR | MSMT | | 78.1 | 72.4 |
| $\text{QAConv}_{50}$ + RR + TLift | MSMT | | 80.3 | 77.2 |
| $\text{QAConv}_{152}$ + RR + TLift | MSMT | | 82.2 | 78.4 |

## 3 Comparisons to Other Losses

Since the loss of hard triplet mining [3] is popular in person re-identification, we further include it in the loss comparisons. Besides, we provide a further analysis on different loss configurations of the QAConv. The results are shown in Table 4 under Market→Duke, where triplet results are each with its best margin. While the mini-batch hard triplet loss does improve the softmax cross-entropy loss, it seems that it is not efficient in learning the QAConv, possibly because local matching requires large pairs to learn, as done with the proposed class memory and focal loss, but not in mini-batches. Note that focal loss is a bit aggressive in learning, but softly. However, the hard triplet loss is in fact more aggressive.

## 4 Fusion of Global Similarity

To see whether fusing a global similarity branch helps improving the performance, we tried an extra global feature learning branch by performing a global average pooling on the final feature maps, and a softmax cross-entropy loss for classification. During testing, the cosine similarity computed from this global feature branch is fused to the QAConv similarity. However, after trying different weights of the two losses, the best mAP we can get is 28.4% under Market→Duke, with the weight 0.001 of the global branch. It is a bit worse than the default

**Table 3.** Comparison (%) of backbone networks with Market-1501 as the target dataset.

| Method | Training | | Test: Market | |
|---|---|---|---|---|
| | Source | Target | R1 | mAP |
| $\text{QAConv}_{50}$ | Duke | | 58.6 | 27.2 |
| $\text{QAConv}_{152}$ | Duke | | 62.8 | 31.6 |
| $\text{QAConv}_{50}$ + RR | Duke | | 65.7 | 45.8 |
| $\text{QAConv}_{152}$ + RR | Duke | | 68.5 | 51.2 |
| $\text{QAConv}_{50}$ + RR + TLift | Duke | | 74.6 | 51.5 |
| $\text{QAConv}_{152}$ + RR + TLift | Duke | | 78.7 | 58.2 |
| $\text{QAConv}_{50}$ | MSMT | | 72.6 | 43.1 |
| $\text{QAConv}_{152}$ | MSMT | | 73.9 | 46.6 |
| $\text{QAConv}_{50}$ + RR | MSMT | | 77.4 | 65.6 |
| $\text{QAConv}_{152}$ + RR | MSMT | | 79.2 | 69.1 |
| $\text{QAConv}_{50}$ + RR + TLift | MSMT | | 86.5 | 72.2 |
| $\text{QAConv}_{152}$ + RR + TLift | MSMT | | 88.4 | 76.0 |

**Table 4.** Role of loss functions under Market→Duke (%).

| Method | | Rank-1 | mAP |
|---|---|---|---|
| ResNet-152 | Softmax cross-entropy | 34.9 | 18.4 |
| | Softmax cross-entropy + triplet | 39.6 | 23.0 |
| | Arc loss [1] | 35.3 | 17.1 |
| | Center loss [5, 4] | 38.9 | 22.1 |
| | Class memory loss | 40.7 | 21.8 |
| $\text{QAConv}_{50}$ | Mini-batch triplet (w/o class memory) | 42.2 | 23.7 |
| | Softmax cross-entropy | 43.4 | 24.9 |
| | Binary cross-entropy | 46.1 | 27.3 |
| | Softmax cross-entropy + triplet | 44.3 | 24.2 |
| | Binary cross-entropy + triplet | 44.7 | 23.6 |
| | Focal loss + triplet | 43.3 | 23.2 |
| | Focal loss (default) | **48.8** | **28.7** |

QAConv (28.7%). This may be because the vanilla global feature branch cannot handle misalignments and occlusions, and so more advanced techniques are needed here. This deserves a further study.

## 5 TLift for Other Methods

Note that TLift can also be generally applied to other methods for improvements. To demonstrate this, Tables 5 and 6 show results of applying TLift to all baseline methods under Market→Duke and Duke→Market, respectively. It can be observed that, beyond the improvements made by re-ranking, TLift can further improve all baseline methods. The improvements are consistently large, with Rank-1 improved by 10.1%-14.1%, and mAP improved by 3.6%-11.1%.

**Table 5.** Role of TLift under Market→Duke (%).

| Method | Original | | + RR | | + RR + TLift | |
|---|---|---|---|---|---|---|
| | Rank-1 | mAP | Rank-1 | mAP | Rank-1 | mAP |
| Softmax cross-entropy | 34.9 | 18.4 | 41.5 | 30.5 | 51.7 | 39.7 |
| Arc loss [1] | 35.3 | 17.1 | 39.8 | 26.3 | 51.0 | 34.8 |
| Center loss [5, 4] | 38.9 | 22.1 | 42.5 | 31.5 | 56.6 | 42.6 |
| Class memory loss | 40.7 | 21.8 | 47.8 | 36.1 | 59.6 | 46.2 |
| QAConv | **54.4** | **33.6** | **61.8** | **52.4** | **70.0** | **61.2** |

**Table 6.** Role of TLift under Duke→Market (%).

| Method | Original | | + RR | | + RR + TLift | |
|---|---|---|---|---|---|---|
| | Rank-1 | mAP | Rank-1 | mAP | Rank-1 | mAP |
| Softmax cross-entropy | 48.5 | 21.4 | 53.2 | 33.7 | 63.3 | 38.0 |
| Arc loss [1] | 48.9 | 21.4 | 54.5 | 34.8 | 64.8 | 39.3 |
| Center loss [5, 4] | 48.8 | 22.0 | 52.5 | 33.3 | 63.0 | 36.9 |
| Class memory loss | 47.8 | 20.5 | 52.9 | 33.1 | 63.4 | 37.5 |
| QAConv | **62.8** | **31.6** | **68.5** | **51.2** | **78.7** | **58.2** |

## 6 Parameter Analysis

Considering the memory consumption and the efficiency, the kernel size of QAConv is set to $s = 1$. Parameters for TLift are $\tau = 100$, $\sigma = 200$, $K = 10$, and $\alpha = 0.2$. They were fixed in all experiments after some initial tries. To understand their influence, we vary them one by one, with corresponding results shown in Tables 7 and 8. It can be observed that, the parameters are not sensitive in a broad range, so that they are easy to select. Besides, some better results can be obtained by varying parameters other than the defaults.

**Table 7.** Influence of TLift parameters under Market→Duke (%). Bold numbers are with the default parameters.

| $\tau$ | 50 | **100** | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 69.3 | **70.0** | 69.7 | 69.8 | 69.1 | 68.3 | 66.8 | 65.5 | 64.4 | 63.9 |
| mAP | 60.7 | **61.2** | 60.7 | 59.9 | 58.8 | 57.3 | 55.7 | 54.0 | 52.4 | 51.2 |

| $\sigma$ | 50 | 100 | 150 | **200** | 250 | 300 | 350 | 400 | 450 | 500 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 67.4 | 69.5 | 70.4 | **70.0** | 69.4 | 69.2 | 68.9 | 68.4 | 68.0 | 67.7 |
| mAP | 55.4 | 59.6 | 60.9 | **61.2** | 61.0 | 60.8 | 60.5 | 60.1 | 59.8 | 59.5 |

| $K$ | 5 | **10** | 15 | 20 | 30 | 40 | 50 | 100 | 150 | 200 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 69.7 | **70.0** | 70.2 | 70.0 | 69.4 | 68.9 | 68.2 | 67.0 | 65.5 | 64.8 |
| mAP | 60.8 | **61.2** | 61.2 | 61.0 | 60.3 | 59.6 | 58.8 | 56.8 | 55.7 | 55.2 |

| $\alpha$ | 0.01 | 0.02 | 0.05 | 0.1 | **0.2** | 0.3 | 0.4 | 0.5 | 0.7 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 70.4 | 70.4 | 70.2 | 70.2 | **70.0** | 69.4 | 69.1 | 68.6 | 68.3 | 67.5 |
| mAP | 60.8 | 60.8 | 60.8 | 61.0 | **61.2** | 61.1 | 61.0 | 60.9 | 60.4 | 59.7 |

**Table 8.** Influence of TLift parameters under Duke→Market (%). Bold numbers are with the default parameters.

| $\tau$ | 50 | **100** | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 76.2 | **78.7** | 79.8 | 79.7 | 79.9 | 79.0 | 78.6 | 78.2 | 77.6 | 77.2 |
| mAP | 57.2 | **58.2** | 58.6 | 58.4 | 58.2 | 57.7 | 57.2 | 56.6 | 56.0 | 55.4 |

| $\sigma$ | 50 | 100 | 150 | **200** | 250 | 300 | 350 | 400 | 450 | 500 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 76.1 | 78.5 | 78.6 | **78.7** | 78.6 | 78.1 | 78.0 | 77.9 | 77.9 | 77.6 |
| mAP | 55.6 | 57.6 | 58.1 | **58.2** | 58.5 | 58.7 | 58.8 | 59.0 | 59.1 | 59.2 |

| $K$ | 5 | **10** | 15 | 20 | 30 | 40 | 50 | 100 | 150 | 200 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 79.6 | **78.7** | 78.1 | 77.6 | 76.6 | 76.2 | 75.8 | 74.4 | 73.4 | 72.7 |
| mAP | 56.9 | **58.2** | 58.4 | 58.3 | 58.0 | 57.9 | 57.8 | 57.3 | 56.6 | 55.9 |

| $\alpha$ | 0.01 | 0.02 | 0.05 | 0.1 | **0.2** | 0.3 | 0.4 | 0.5 | 0.7 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| Rank-1 | 78.4 | 78.5 | 78.7 | 78.8 | **78.7** | 78.5 | 78.0 | 77.6 | 76.5 | 75.4 |
| mAP | 53.8 | 54.1 | 55.0 | 56.3 | **58.2** | 59.4 | 59.9 | 60.0 | 59.5 | 58.5 |

## 7 Memory Usage

One drawback of QAConv is that it requires more memory to run than other methods, and it needs to store feature maps of images, rather than features, where feature maps are generally larger in size than representation features. For training on the DukeMTMC-reID, the GPU memory consumption for the QAConv is about 2.83GB, while that for the softmax baseline is about 2.78GB. They are comparable because though QAConv spends some more on class memory, it uses three layers of the ResNet-50, while the softmax baseline uses four layers. For inference, the peak GPU memory for the QAConv is about 2.3GB, while that for the softmax baseline is about 1.7GB.

## Biography

Shengcai Liao is a Lead Scientist in the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, UAE. He is a Senior Member of IEEE. Previously, he was an Associate Professor in the Institute of Automation, Chinese Academy of Sciences (CASIA). He received the B.S. degree in mathematics from the Sun Yat-sen University in 2005 and the Ph.D. degree from CASIA in 2010. He was a Postdoc in the Michigan State University during 2010-2012. His research interests include object detection, face recognition, and person re-identification. He has published over 100 papers, with over 11,000 citations according to Google Scholar. He was awarded the Best Student Paper in ICB 2006, ICB 2015, and CCBR 2016, and the Best Paper in ICB 2007. He was also awarded the IJCB 2014 Best Reviewer and CVPR 2019 Outstanding Reviewer. He was an Assistant Editor for the book "Encyclopedia of Biometrics (2nd Ed.)". He also served as Area Chairs for ICPR 2016, ICB 2016, and ICB 2018, and reviewers for ICCV, CVPR, ECCV, TPAMI, IJCV, TIP, TIFS, etc. He was the Winner of the CVPR 2017 Detection in Crowded Scenes Challenge and the ICCV 2019 NightOwls Pedestrian Detection Challenge. Homepage: https://liaosc.wordpress.com/

Ling Shao (Senior Member, IEEE) is currently the Executive Vice President and a Provost of the Mohamed bin Zayed University of Artificial Intelligence. He is also the CEO and the Chief Scientist of the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. His research interests include computer vision, machine learning, and medical imaging. He is a fellow of IAPR, IET, and BCS.

## References

1. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)

2. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
3. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
4. Jin, H., Wang, X., Liao, S., Li, S.Z.: Deep person re-identification with improved embedding and efficient training. In: 2017 IEEE International Joint Conference on Biometrics (IJCB). pp. 261–267. IEEE (2017)
5. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: European conference on computer vision. pp. 499–515. Springer (2016)
6. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI) (2020)