Gallery Filter Network for Person Search
Supplementary Material

Lucas Jaffe
UC Berkeley
lucasjaffe@berkeley.edu

Avideh Zakhor
UC Berkeley
avz@berkeley.edu

A. Data Processing and Evaluation

We make publicly available our codebase[1], which includes instructions and config files needed to replicate all main experiments of the paper. For comparitive purposes, we implicitly refer in the following subsections to the public codebases of OIM[2] [8], NAE[3] [3], SeqNet[4] [4], COAT[5] [10], AlignPS[6] [9], and PSTR[7] [2].

A.1. Standardized Data Format

We produce an intermediate COCO-style [6] format for all partitions of the CUHK-SYSU and PRW datatsets. In addition to standard COCO object metadata, we include `person_id` and `is_known` fields for persons, and a `cam_id` image field for performing cross-camera evaluation.

This standardization process made it straightforward to prepare new partitions of the data. In particular, we split the standard training sets into separate training and validation sets, and created some additional smaller debugging sets. This allowed us to pick hyperparameters without fitting to the test data.

We also standardize the format of retrieval partitions into three categories: 1) fully-specified format which encodes the exact gallery scenes to be used for each query 2) format which specifies queries only, and uses all scenes in the partition as the gallery and 3) format which uses all possible queries, and all possible scenes as the gallery. We create the second and third formats because it is otherwise inefficient to fully-specify the "all" cases.

A.2. Training and Validation Sets

For both datasets, known identity sets between the train and test partitions are disjoint, making the standard evaluation an *open-set* retrieval problem. To construct the training and validation sets to mirror the open-set retrieval problem of the standard train-test divide, we build a graph based on which scenes share common person identities. Two nodes (scenes) have an edge between them if they share at least one person identity in common. In this way, we can easily split the CUHK-SYSU dataset into a set of connected components, and divide those components into two groups for train (∼80%) and val (∼20%).

Since the PRW dataset comprises video surveillance footage, this graph has the property that nearly every scene is connected to another scene via some common person identity. Therefore, we ignore the top 100 most common person identities when constructing the graph for PRW, resulting in a partition which is not quite open-set, but should exhibit similar generalization properties for the purpose of model development. For PRW, we also divide components into two groups for train (∼80%) and val (∼20%).

We rename the original train set to "trainval", and all of our final experimental results in this paper are from models trained on the full trainval set, and tested on the full test set using the standard retrieval scenarios.

	CUHK-SYSU				PRW			
Metadata	trainval	test	train	val	trainval	test	train	val
Scenes	11,206	6,978	8,964	2,242	5,704	6,112	4,563	1,141
Boxes	55,272	40,871	44,244	11,028	18,048	25,062	14,897	3,151
Known IDs	5,532	2,900	4,296	1,236	483	544	424	158
Known Boxes	15,085	8,345	11,623	3,462	14,907	19,127	12,125	2,782
Unknown Boxes	40,187	32,526	32,621	7,566	3,141	5,935	2,772	369

Table 1: Dataset metadata showing how many scenes, boxes, and person IDs are in each partition.

A.3. Partition Information

Metadata for the exact breakdown of known and unknown identities and boxes for each partition is given in Table 1.

[1] https://github.com/LukeJaffe/GFN
[2] https://github.com/ShuangLI59/person_search
[3] https://github.com/dichen-cd/NAE4PS
[4] https://github.com/serend1p1ty/SeqNet
[5] https://github.com/Kitware/COAT
[6] https://github.com/daodaofr/AlignPS
[7] https://github.com/JialeCao001/PSTR

A.4. Evaluation Functions

Using these standardized partitions, we are able to use just one function for detection evaluation and one for retrieval evaluation, as opposed to separate functions for each dataset. This also makes it easier to add in method-specific metrics that can be immediately tested for all partitions.

We note that the current dataset releases for PRW and CUHK-SYSU have a small number (5 or less) of the following errors: duplicate bounding boxes in a single scene, repeated person ids in a single scene, and repeated gallery scenes in a retrieval partition. Although these issues are not handled correctly by the standard evaluation function, we exactly replicate the previous erroneous behavior in our new evaluation function to be certain the comparison against other methods is fair. We leave correction of the underlying data and evaluation function to future work.

A.5. Augmentation Code Structure

To make use of augmentation strategies in the albumentations library [1], we refactor evaluation to occur on the augmented data instead of the original data. This allows for easy inclusion of different resizing and cropping strategies which we make use of, in addition to a wealth of other augmentations, experimenting with which we leave to future work.

A.6. Config Format and Ray Tune

For running experiments with our code, we provide a YAML config format which is compatible with the Ray Tune library [5]. We specifically support the `tune.grid_search` functionality by parsing lists in the YAML file as inputs to this function. This makes it easy to run ablations with many variations using a single config file.

B. Additional Implementation Details

Model Details. We set the OIM scalar (inverse temperature) parameter to 30.0 as in [4], with an OIM circular queue size of 5,000 for CUHK-SYSU and 500 for PRW. The OIM momentum parameter is also left at 0.5. For the GFN, the training temperature parameter is 0.1, and the GFN excitation function temperature parameter is 0.2. During training, we use a batch size of 12 for ResNet50 backbone models, and a batch size of 8 for ConvNeXt backbone models.

For the ResNet50 backbone, we freeze all batch norm layers, and all weights through the `conv1` layer of the model. For ConvNeXt backbones, we freeze only the `conv1` layer of the model. All backbones are initialized using weights from pre-training on ImageNet1k [7].

We use automatic-mixed precision (AMP), which significantly reduces all training and inference times. To avoid `float16` overflow, we refactor all loss functions to divide before summation when computing mean reduction. This increases likelihood of underflow, but results in more stable training overall.

	Detection		Re-id				GFN	
Model	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
RN50 NAE-FCS	97.6	93.5	50.3	89.4	0.0	+3.5	16.7	78.5
RN50 RCNN-SCS	96.0	93.1	51.1	90.6	+0.1	**+3.8**	16.3	78.0
CNB NAE-FCS	**97.9**	**94.9**	58.7	91.4	**+1.3**	+3.4	**21.0**	**78.9**
CNB RCNN-SCS†	96.0	93.4	**58.8**	**92.3**	+1.1	+3.5	20.4	78.5

Table 2: Comparison of model backbone (RN50=ResNet50, CNB=ConvNeXt Base), NAE vs. R-CNN head for the second detector stage, and first (stage) classifier score (FCS) or second (stage) classifier score (SCS) used at inference time. Baseline model is marked with †, final model is highlighted gray.

GFN Sampling Strategies. Since we are unable to use the entire GFN LUT to form loss pairs in any given batch due to memory limitations, we have a choice about which LUT embeddings to select for the GFN optimization. By default, for each query person present in the current batch, we sample one matching scene embedding and the person embeddings for all persons in that scene. In addition, we consider sampling a "hard negative" scene, defined as a scene which shares at least one person identity in common with the query scene, but that does not contain the query person identity. An ablation for related choices is considered in Section D.

C. Qualitative Analysis

Qualitative examples are shown for both CUHK-SYSU and PRW in Figure 1. All examples show cases where the baseline model top-1 match is incorrect, but the GFN-modified match for the same example is correct. We highlight examples where global scene context has an obvious vs. a more subtle impact, and where the query and scene camera ID are the same or different.

D. Additional Ablations

Model Modifications. We consider how changes to the SeqNet architecture impact performance, including usage of a second Faster R-CNN head instead of the NAE head, and usage of the second detector stage score instead of the first stage score during inference. Results are shown in Table 2.

Using the ConvNeXt Base backbone instead of ResNet50 does not improve detection performance, but it significantly improves re-id performance, especially mAP, by 7-8%. Using the first stage score significantly helps detection performance, but it reduces re-id performance.

Image Augmentation. Shown in Table 3, we compare the Window Resize augmentation to the two cropping methods used, and a strategy combining the two. We find that the Window Resize method achieves comparable re-id performance with other methods, but much lower detection performance. This may be attributed to the regularizing effect

Figure 1: Retrieval examples (CUHK-SYSU left, PRW right) from the baseline model where application of the GFN score corrected the top-1 result. The query box is shown in yellow, a false positive gallery match in red, and a true positive gallery match in blue. In each scene, the white box in the lower right duplicates the person of interest for easier comparison between scenes. In the top-left and middle-left, subtle contextual clues (formal wear) help correct the predicted box. In the bottom-left, an obvious contextual clue (interior of same building) corrects the prediction, despite a 180° change in viewpoint of the person. In the top-right, the false positive and correct match look nearly identical, and the correct box is from the same camera view. In the middle-right, the false positive and correct match have the same shirt and hairstyle, and the correct box is from a different camera view. In the lower-right, the false positive appears to be a mistake in the ground truth (should be true positive), but the GFN "helped" by up-weighting a more contextually similar scene.

	Detection		Re-id				GFN	
Method	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
WRS	89.3	87.7	57.3	91.1	+0.9	**+4.7**	19.6	78.3
RSC	95.9	93.1	55.8	91.0	+0.7	+3.7	18.5	77.6
RFC	95.0	92.7	58.4	91.2	**+1.4**	+3.4	20.8	77.8
RFC2	95.4	93.1	58.2	91.1	**+1.4**	+3.8	**21.1**	78.4
RSC+RFC†	96.0	93.4	**58.8**	**92.3**	+1.1	+3.5	20.4	78.5
RSC+RFC2	**96.1**	**93.8**	58.7	**92.3**	+1.3	+3.3	20.8	**78.9**
Crop Size	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
256×256	95.3	91.9	51.4	90.1	0.1	3.3	16.7	78.0
384×384	**96.3**	**93.6**	56.7	92.0	0.6	3.0	19.6	79.2
512×512†	96.0	93.4	58.8	**92.3**	1.1	**3.5**	20.4	78.5
640×640	95.3	92.9	**59.6**	**92.3**	**1.4**	3.4	**21.8**	**79.6**

Table 3: Comparison of image augmentation methods (top), and image crop sizes (bottom). Augmentation methods include WRS (Window Resize to 900×1500), RSC (Random Safe Crop to square crop size), RFC (Random Focused Crop to square crop size), RFC2 (variant of RFC), and RSC+RFC(2) which performs either cropping method randomly with equal probability. Baseline model is marked with †, final model is highlighted gray.

of random cropping for detector training.

In addition, we find that Random Safe Cropping alone results in better detection performance than Random Focused Cropping alone, but worse re-id performance. This shows that the regularizing effect of random crops that may be in the wrong scale is more important for detection, and having features in the target scene scale is more important for re-id. Combining the two results in better performance than either alone for both detection and re-id.

Scene Pooling Size and Embedding Dimension. We analyze choices for the RoI Align pooling size for the scene embedding head, and choices for the embedding dimension for both the query and scene embedding heads. Comparisons are shown in Table 4.

GFN performance increases nearly-monotonically with scene pooling size, with diminishing returns for GFN score-weighted re-id performance. We also note that larger scene pooling size results in a significant increase in memory consumption, so we use 56×56 by default, which captures most of the performance gain, with some memory savings.

It is clear that the scene pooling size should be larger than the query pooling size to ensure that all person information in a scene is adequately captured. The relationship between person box size distribution vs. scene size, with the ratio of respective pooling sizes could be further investigated.

For the embedding dimension, performance also increases nearly-monotonically with size, for both re-id and the GFN-only stats. Although there are diminishing returns in performance, like with the scene pooling size, we choose the relatively large value of 2,048 because it results in little additional memory consumption or compute time.

GFN Sampling. We analyze choices for the GFN sampling procedure, with comparisons shown in Table 5. Critically, we find that all sampling options with the LUT are better than not using the LUT at all, as shown by both the large increase in GFN stats, and the contribution of GFN score-weighting to re-id stats. This is expected but important, because it shows that batch-only query-scene comparisons are insufficient (usually just comparing a query to the scene it is present in), and that LUT comparisons are needed despite no gradients flowing through the LUT.

Among sampling mechanisms that use the LUT, results for GFN score-weighted re-id stats were relatively similar,

	Detection		Re-id				GFN	
Pool Size	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
14×14	**96.1**	93.5	58.1	91.6	+0.1	+3.3	18.2	77.9
28×28	95.9	93.4	58.5	92.3	+0.7	**+3.6**	19.7	79.2
56×56†	96.0	93.4	**58.8**	92.3	+1.1	+3.5	20.4	78.5
112×112	**96.1**	**93.6**	**58.8**	**92.4**	**+1.2**	**+3.6**	**22.1**	**79.8**
Emb Dim	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
128	96.1	**93.6**	58.0	91.6	0.7	3.8	19.6	77.9
256	95.9	93.4	58.2	92.0	1.0	**4.3**	20.1	78.3
512	96.1	93.5	58.7	91.8	1.0	4.0	20.0	77.6
1024	**96.2**	**93.6**	**59.3**	92.2	**1.1**	3.5	20.0	78.0
2048†	96.0	93.4	58.8	**92.3**	**1.1**	3.5	**20.4**	**78.5**

Table 4: Comparison of pooling sizes for the RoI Align block used to compute scene embeddings (top) and comparison of the embedding dimension used for both query and scene embeddings (bottom). Baseline model is marked with †, final model is highlighted gray.

	Detection		Re-id				GFN	
Sampling	Recall	AP	mAP	top-1	Δ mAP	Δ top-1	mAP	top-1
No LUT	**96.2**	**93.7**	57.5	90.8	-0.3	+2.1	13.3	72.8
P1N0	96.1	93.6	**59.5**	91.9	**+1.3**	+2.4	21.0	78.7
P1N1†	96.0	93.4	58.8	**92.3**	+1.1	+3.5	20.4	78.5
P2N0	**96.2**	**93.7**	59.1	91.9	+1.2	**+3.6**	20.9	**79.5**
P2N1	96.0	93.6	59.1	91.7	+1.2	+3.4	**21.1**	**79.5**

Table 5: Comparison of different sampling options for optimization of the GFN. PxNy indicates that x positive scenes and y hard negative scenes are sampled for each person in the batch. No LUT means we use only batch query and scene embeddings, and no LUT is used. Baseline model is marked with †, final model is highlighted gray.

and more trials with more samples per trial are likely needed to distinguish a standout method.

E. Comparison with CBGM

The GFN module is similar to the Context Bipartite Graph Matching (CBGM) method from [4] in that both methods use context from the query and gallery scenes to improve prediction ranking, although the GFN is used at inference-time only, and does not need to be trained. CBGM is more explicit, in that it directly attempts to match detected person boxes in the query and gallery scenes, at the expense of requiring sensitive hyperparameters: the number of boxes to use from each scene for the matching. The authors found that very different values for these parameters were optimal for the CUHK-SYSU vs. PRW datasets, and did not provide a clear methodology for their selection besides test set performance. In contrast, we use the exact same GFN configuration for both datasets during training and inference, selected separately based on validation data, and found it to robustly improve performance for both.

References

[1] Alexander Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, and Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, Feb. 2020. arXiv:1809.06839 [cs].

[2] Jiale Cao, Yanwei Pang, Rao Muhammad Anwer, Hisham Cholakkal, Jin Xie, Mubarak Shah, and Fahad Shahbaz Khan. PSTR: End-to-End One-Step Person Search With Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9458–9467, 2022.

[3] Di Chen, Shanshan Zhang, Jian Yang, and Bernt Schiele. Norm-Aware Embedding for Efficient Person Search. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12612–12621, Seattle, WA, USA, June 2020. IEEE.

[4] Zhengjia Li and Duoqian Miao. Sequential End-to-end Network for Efficient Person Search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):2011–2019, May 2021. Number: 3.

[5] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A Research Platform for Distributed Model Selection and Training, July 2018. arXiv:1807.05118 [cs, stat].

[6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision ECCV 2014*, Lecture Notes in Computer Science, pages 740–755, Cham, 2014. Springer International Publishing.

[7] 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, Dec. 2015.

[8] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint Detection and Identification Feature Learning for Person Search. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3376–3385, Honolulu, HI, July 2017. IEEE.

[9] Yichao Yan, Jinpeng Li, Jie Qin, Song Bai, Shengcai Liao, Li Liu, Fan Zhu, and Ling Shao. Anchor-Free Person Search. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7686–7695, June 2021. ISSN: 2575-7075.

[10] Rui Yu, Dawei Du, Rodney LaLonde, Daniel Davila, Christopher Funk, Anthony Hoogs, and Brian Clipp. Cascade Transformers for End-to-End Person Search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7267–7276, 2022.