Representation Learning for Person Re-Identification: Baseline Reproduction and Variants on Market-1501 and DukeMTMC-reID

Ruiheng Li
FST, University of Macau, Macau, China
Macau, China
mc565072@um.edu.mo

ABSTRACT

Person re-identification (ReID) is a representation learning problem where a model must match the same identity across cameras under large appearance changes. In this report, we reproduce a strong baseline on Market-1501 and study variants on DukeMTMC-reID using the public PyTorch ReID baseline code. On Market-1501, the ResNet-50 baseline achieves Rank@1 = 0.8774 and mAP = 0.7219. We also observe a severe domain shift when directly testing the Market-trained model on DukeMTMC-reID (Rank@1 = 0.3299, mAP = 0.1700). Training on DukeMTMC-reID recovers performance (ResNet-50 Rank@1 = 0.7935, mAP = 0.6174), and using a stronger backbone (HRNet) further improves results (Rank@1 = 0.8389, mAP = 0.6946). Finally, we analyze two concrete failure cases and discuss improvement directions and AI safety considerations.

KEYWORDS

Person Re-identification, Representation Learning, Metric Learning, Retrieval, Deep Learning

1 INTRODUCTION

Person ReID aims to retrieve images of the same person identity from a gallery given a query image. Compared with closed-set classification, ReID is typically evaluated as a retrieval problem using embedding representations and distance-based ranking. This makes ReID a practical lens for studying representation learning, because good embeddings must be discriminative (separate different identities) and robust (invariant to camera, pose, occlusion, and illumination).

In this work, we reproduce a baseline model on Market-1501 [3] and run controlled variants on DukeMTMC-reID, covering baseline reproduction, cross-dataset domain shift, the impact of different backbones and loss functions, and failure-case analysis with safety reflection.

2 METHOD

2.1 Baseline architecture

We follow the codebase default design: a CNN backbone (ResNet-50 [1] by default) produces a feature map, followed by global pooling to obtain a fixed-length descriptor. The descriptor is projected to a 512-dimensional embedding and trained with an identity classification head (softmax cross-entropy).

2.2 Testing protocol

At test time, query and gallery embeddings are extracted and matched by distance. The evaluation reports Cumulative Matching Characteristics (CMC) and mean Average Precision (mAP). We also use a common test-time augmentation: horizontal flip, where features from the original and flipped image are averaged.

3 EXPERIMENTAL SETUP

3.1 Datasets

Market-1501 [3] is a widely-used ReID benchmark with 751 training identities and evaluation by query/gallery split. **DukeMTMC-reID** [2, 4] is another benchmark with 702 training identities and similar evaluation protocol.

3.2 Training configuration

Unless otherwise stated, we use the same training recipe across experiments for fair comparison: batch size 32, total epochs 60, initial learning rate 0.05, weight decay 5×10^{-4}, and dropout 0.5. The code is configured to train with $-$train_all, which removes a held-out validation split for hyper-parameter tuning. This simplifies the pipeline but increases the risk of overfitting and makes model selection less principled; therefore, the final comparison relies on the fixed query/gallery evaluation.

3.3 Commands and reproducibility

We use the following commands:

```
python train.py --gpu_ids 0 --name ft_ResNet50 --train_all
      --batchsize 32 --data_dir
      ./data/Market-1501-v15.09.15/pytorch
python test.py --gpu_ids 0 --name ft_ResNet50 --test_dir
      ./data/Market-1501-v15.09.15/pytorch --batchsize 32
      --which_epoch 060
python train.py --gpu_ids 0 --name duke_ft_ResNet50 --train_all
      --batchsize 32 --data_dir ./data/DukeMTMC-reID/pytorch
python train.py --gpu_ids 0 --name duke_ft_HR --train_all
      --batchsize 32 --data_dir ./data/DukeMTMC-reID/pytorch
      --use_hr
```

4 RESULTS

4.1 Market-1501 baseline reproduction

Table 1 summarizes the reproduced baseline results on Market-1501.

Model	Rank@1	Rank@5	Rank@10	mAP
ResNet-50 (softmax)	0.8774	0.9561	0.9724	0.7219

Table 1: Baseline results on Market-1501 using ft_ResNet50 **(epoch 60).**

Setting	Rank@1	Rank@5	Rank@10	mAP
Market → Duke (zero-shot)	0.3299	0.4852	0.5485	0.1700
Duke baseline (ResNet-50)	0.7935	0.8896	0.9201	0.6174
Duke + DenseNet (–use_dense)	0.8151	0.9129	0.9367	0.6484
Duke + HRNet (–use_hr)	**0.8389**	**0.9241**	**0.9421**	**0.6946**
Duke + Circle loss (–circle)	0.7828	0.8842	0.9125	0.6128
Duke + Instance loss (–instance)	0.8007	0.8896	0.9183	0.6232
Duke + Triplet loss (–triplet)	0.7931	0.8878	0.9215	0.6232

Table 2: Results on DukeMTMC-reID (epoch 60). "Market → Duke" denotes testing Duke using the Market-trained ft_ResNet50 **without fine-tuning.**

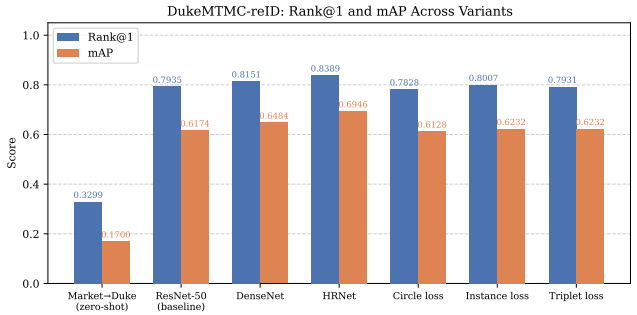

Figure 1: Rank@1 and mAP comparison across DukeMTMC-reID variants.

4.2 Domain shift: Market-trained model on Duke

Directly applying the Market-trained model on Duke results in a large performance drop (Rank@1 = 0.3299, mAP = 0.1700), indicating strong domain shift. This is consistent with the intuition that camera networks, backgrounds, illumination, and clothing distribution differ across datasets, so representations trained for one domain may not transfer without adaptation.

4.3 Duke training and variants

Table 2 reports Duke results when training on Duke as well as variants. Among tested backbones, HRNet yields the best performance in our runs.

5 QUICK QUESTIONS

This section summarizes key practical questions from the tutorial and our experiments.

Why use AdaptiveAvgPool2d? Adaptive average pooling (e.g., output size 1×1) produces a fixed spatial output regardless of the input resolution. This keeps the head shape-stable, while fixed-kernel average pooling would make the output depend on the input size.

Why horizontally flip images in test? Flip test-time augmentation averages features from original and flipped images. This reduces sensitivity to left-right pose bias and often improves retrieval robustness with minimal compute overhead.

Why L2-normalize features? L2-normalization puts embeddings on the unit hypersphere. Distances then emphasize angular differences and become less sensitive to scale, making similarity computation more stable for retrieval.

Why call optimizer.zero_grad()**?** Gradients accumulate by default in PyTorch. Without clearing old gradients each iteration, updates implicitly sum gradients across steps, which usually destabilizes training unless gradient accumulation is intended.

Figure 2: Two DukeMTMC-reID failure cases (Top-10 retrieval visualization). Top: Case-1 (query index 1883, ID 4315). Bottom: Case-2 (query index 67, ID 0051).

6 FAILURE CASE ANALYSIS

We analyze two failure cases from Duke evaluation (visualizations in Figure 1). Both cases exhibit a "no positives in Top-10" pattern, where the first correct match appears extremely late in the ranking.

6.1 Case-1: missed match under cross-camera variation

Query image: 4315_c6_f0076814.jpg (ID 4315, query index 1883). The Top-1 retrieved image is an incorrect identity, 6367_c8_f0073570.jpg, and the first true match does not appear until rank 4274 (4315_c7_f0080881.jpg). This large rank gap indicates weak cross-camera invariance: viewpoint and illumination changes likely shift the query embedding away from its true cluster, while visually similar hard negatives occupy the top positions.

6.2 Case-2: false positives dominate Top-K

Query image: 0051_c1_f0060060.jpg (ID 0051, query index 67). The Top-1 result is also a false positive (6794_c8_f0178831.jpg), and the first correct gallery image appears only at rank 3277 (0051_c2_f0060553.jpg). The pattern suggests that the model over-relies on non-identity cues, such as background context or coarse clothing color, producing confident but semantically incorrect matches in early retrieval ranks.

7 IMPROVEMENT PROPOSALS

Based on the above results and observed failures, we propose:

- **Re-ranking:** apply k-reciprocal re-ranking or GNN-based re-ranking to refine the initial ranking. This often improves mAP by correcting hard negatives at high ranks, at the cost of additional time and memory.
- **Stronger augmentation and sampling:** combine random erasing, color jitter, and identity-balanced sampling to improve robustness to occlusion and illumination changes, reducing overfitting to dataset-specific cues.

8 AI SAFETY REFLECTION

ReID is a dual-use technology. It can support positive applications such as search-and-rescue and authorized security auditing, but it can also enable non-consensual tracking and discriminatory surveillance. We suggest safeguards aligned with "lawful authorization + minimal necessity + auditability": minimize retention (store only necessary encrypted embeddings), enforce access control and audit logs, apply rate limiting and thresholding for external queries, and conduct governance reviews for high-risk deployments.

9 CONCLUSION

We reproduced a strong ReID baseline on Market-1501 and evaluated transfer and variants on DukeMTMC-reID. The reproduced Market baseline achieves Rank@1 = 0.8774 and mAP = 0.7219. Zero-shot transfer from Market to Duke fails badly, highlighting domain shift. Training on Duke restores performance, and a stronger backbone (HRNet) improves retrieval quality further. Failure-case analysis suggests that cross-camera variation and hard negatives remain major challenges, motivating re-ranking and stronger augmentation as practical next steps.

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[2] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. 2016. Performance measures and a data set for multi-target, multi-camera tracking. In *European conference on computer vision*. Springer, 17–35.

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

[4] Zhedong Zheng, Liang Zheng, and Yi Yang. 2017. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proceedings of the IEEE international conference on computer vision*. 3754–3762.