UTS UNIVERSITY OF TECHNOLOGY SYDNEY

# Generative and Discriminative Learning for Visual Matching

**by Zhedong Zheng**

Thesis submitted in fulfilment of the requirements for the degree of

**Doctor of Philosophy**

under the supervision of Prof. Yi Yang and Dr. Liang Zheng

University of Technology Sydney
Faculty of Engineering and Information Technology

May 2021

# CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Zhedong Zheng, declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science at the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note:
Signature removed prior to publication.

DATE: 24, May, 2021

# ABSTRACT

**Generative and Discriminative Learning**

**for Visual Matching**

by

Zhedong Zheng

Visual matching aims to establish image correspondences across viewpoints. Given a query image, the visual matching system seeks to retrieve images containing the object of interest from non-overlapping viewpoints according to the similarity score. The visual matching task remains challenging because objects captured by different viewpoints often contain significant intra-class variations caused by background, viewpoint, object pose, etc. In this thesis, I present my research on combining generative learning with discriminative learning to build one robust visual matching system. First, due to lack of sufficient data to enhance robustness against input variations, generative learning is aimed at letting the model potentially "see" these variations (particularly intra-class variations) during training. With recent progress in the generative adversarial networks (GANs), generative models have become appealing choices to introduce additional augmented data for free. Second, discriminative learning is designed to formulate visual matching as a metric learning problem and adopt the discriminative optimization objective to learn the distance. With these objectives in mind, it motivates us to enable Convolutional Neural Network (CNN) to learn the mapping function to discriminate between different objects. In this thesis, I investigate two scientific problems of combining two learning strategies: 1) How to obtain high-quality generated data for subsequential training? 2) How to leverage the generated data to promote discriminative learning? To study the two problems, I explore improving learned visual representations by better leveraging the data from the following three aspects.

First, we present a semi-supervised pipeline that integrates GAN-generated images into discriminative learning. It contains a generative adversarial model for unsupervised data generation and a discriminative convolutional neural network for semi-supervised learning. Second, we observe that the generative pipelines are typically presented as standalone models, which are relatively separate from the discriminative learning models. To make the best of the two worlds, we further propose a learning framework that couples discriminative and generative learning. This design leads to a unified framework that enables the interactions between generative and discriminative modules in an end-to-end manner. Third, we further investigate different discriminative learning approaches on various data sources. Specifically, we study the feasibility of borrowing the knowledge from real-world vehicle images collected on the web and propose a two-stage learning strategy to minimize the domain gap between the web data and real-world data. Furthermore, we also explore the possibility of learning from synthetic data simulated by 3D engines. We propose a new geo-localization benchmark and build a strong and flexible baseline to learn from multi-view multi-source data.

In summary, this thesis studies and solves the critical challenges of data limitation and robust representation learning in visual matching. We show the benefits of leveraging the generative and discriminative learning in deep learning, which achieves better performance than previous methods.

Dissertation directed by Professor Yi Yang
The Australian Artificial Intelligence Institute (AAII), School of Computer Science

# Acknowledgements

First, I would like to thank my supervisor, Prof. Yi Yang, for his encouragement, guidance, and patience. He has provided his perpetual supports to help me overcome difficulties and guided me to pursue challenging problems. He also taught me how to re-consider the same thing from different views, which is valuable to my future career. I have learned a lot under his caring supervision, and he has my deepest gratitude. Besides, I would like to thank my co-supervisor, Dr. Liang Zheng. He has provided me invaluable suggestions and consistent supports throughout my Ph.D. life. Thanks to Xiaodong Yang, Zhiding Yu, and Jan Kautz, my mentors when I interned at Nvidia Research, from whom I learned critical thinking and time management. Thanks to Mingyue Jiang and Xiao Tan, my mentors when I remotely interned at Baidu Research, from whom I learned the competition skills and knowledge. I also would like to thank all colleagues in the ReLER Lab at the University of Technology Sydney for their friendship and supports. I was fortunate to work with them.

Many thanks to Data to Decision CRC for supporting my research.

Finally, I would like to thank my parents Yanmin Yang and Gaoxin Zheng, my grandma Wenlan Tang for loves and supports over the years.

Zhedong Zheng

Sydney, Australia, 2021.

# List of Publications

**Journal Papers**

J-1. **Zhedong Zheng**, Liang Zheng, Yi Yang (2018). A discriminatively learned cnn embedding for person reidentification. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14(1), 1-20. [**ESI Highly-cited Paper**]

J-2. **Zhedong Zheng**, Liang Zheng, Yi Yang (2018). Pedestrian alignment network for large-scale person re-identification. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), 29(10), 3037-3045. [**ESI Highly-cited Paper**]

J-3. **Zhedong Zheng**, Liang Zheng, Michael Garrett, Yi Yang, Mingliang Xu, Yi-Dong Shen (2020). Dual-Path Convolutional Image-Text Embeddings with Instance Loss. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 16(2), 1-23.

J-4. **Zhedong Zheng**, Tao Ruan, Yunchao Wei, Yi Yang, Mei Tao (2020). VehicleNet: Learning Robust Visual Representation for Vehicle Re-identification. IEEE Transactions on Multimedia (TMM).

J-5. **Zhedong Zheng**, Yi Yang (2021). Rectifying Pseudo Label Learning via Uncertainty Estimation for Domain Adaptive Semantic Segmentation. International Journal of Computer Vision (IJCV), 1-15.

J-6. Zhun Zhong, Liang Zheng, **Zhedong Zheng**, Shaozi Li, Yi Yang (2018). Camstyle: A novel data augmentation method for person re-identification. IEEE Transactions on Image Processing (TIP), 28(3), 1176-1190. [**ESI Highly-cited Paper**]

J-7. Yan Huang, Jinsong Xu, Qiang Wu, **Zhedong Zheng**, Zhaoxiang Zhang, Jian

Zhang (2018). Multi-pseudo regularized label for generated data in person re-identification. IEEE Transactions on Image Processing (TIP), 28(3), 1391-1403.

J-8. Yutian Lin, Liang Zheng, **Zhedong Zheng**, Yu Wu, Zhilan Hu, Chenggang Yan, Yi Yang (2019). Improving person re-identification by attribute and identity learning. Pattern Recognition, 95, 151-161. [**ESI Highly-cited Paper**]

J-9. Qingji Guan, Yaping Huang, Zhun Zhong, **Zhedong Zheng**, Liang Zheng, Yi Yang (2020). Thorax disease classification with attention guided convolutional neural network. Pattern Recognition Letters, 131, 38-45.

J-10. Yutian Lin, **Zhedong Zheng**, Hong Zhang, Chenqiang Gao, Yi Yang (2020). Bayesian query expansion for multi-camera person re-identification. Pattern Recognition Letters, 130, 284-292.

J-11. Bingwen Hu, **Zhedong Zheng**, Ping Liu, Wankou Yang, Mingwu Ren (2020). Unsupervised eyeglasses removal in the wild. IEEE Transactions on Cybernetics, 1 - 13.

J-12. Tingyu Wang, **Zhedong Zheng**, Chenggang Yan, Jiyong Zhang, Yaoqi Sun, Bolun Zheng, Yi Yang (2021). Each Part Matters: Local Patterns Facilitate Cross-view Geo-localization. IEEE Transactions on Circuits and Systems for Video Technology (TCSVT).

**Conference Papers**

C-1. **Zhedong Zheng**, Liang Zheng, 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 (ICCV) (pp. 3754-3762). [Spotlight]

C-2. **Zhedong Zheng**, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, Jan Kautz (2019). Joint discriminative and generative learning for person re-identification. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 2138-2147). [Oral]

C-3. **Zhedong Zheng**, Yunchao Wei, Yi Yang (2020). University-1652: A Multi-view Multi-source Benchmark for Drone-based Geo-localization. In Proceedings of the ACM Multmieda (ACM MM) (pp. 1395–1403).

C-4. **Zhedong Zheng**, Yi Yang (2020). Unsupervised Scene Adaptation with Memory Regularization in vivo. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (pp. 1076-1082).

C-5. Yawei Luo, **Zhedong Zheng**, Liang Zheng, Tao Guan, Junqing Yu, Yi Yang (2018). Macro-micro adversarial network for human parsing. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 418-434).

C-6. Zhun Zhong, Liang Zheng, **Zhedong Zheng**, Shaozi Li, Yi Yang (2018). Camera style adaptation for person re-identification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 5157-5166).

C-7. Zhikun Huang, **Zhedong Zheng**, Chenggang Yan, Hongtao Xie, Yaoqi Sun, Jianzhong Wang, Jiyong Zhang (2020). Real-World Automatic Makeup via Identity Preservation Makeup Net. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (pp. 652-658).

C-8. Chuchu Han, **Zhedong Zheng**, Changxin Gao, Nong Sang, Yi Yang (2021). Decoupled and Memory-Reinforced Networks: Towards Effective Feature Learning for One-Step Person Search. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI).

The source codes of most works can be found at `https://github.com/layumi` .

# Contents

Certificate ii

Abstract iii

Acknowledgments v

List of Publications vi

List of Figures xiii

**1 Introduction 1**

1.1 Motivation . . . . . 2

1.2 Approach . . . . . 2

1.3 Contributions . . . . . 3

1.4 Outline . . . . . 5

**2 Literature Review 7**

2.1 Discriminative Learning . . . . . 7

2.1.1 Person Re-identification . . . . . 7

2.1.2 Vehicle Re-identification . . . . . 8

2.1.3 Cross-view Geo-localization . . . . . 9

2.2 Generative Learning . . . . . 10

2.2.1 Generative Adversarial Networks . . . . . 10

2.2.2 Dataset Augmentation . . . . . 11

2.3 Related Works on Machine Learning . . . . . 12

2.3.1 Semi-supervised Learning . . . . . 12
2.3.2 Transfer Learning . . . . . 13

**3 Semi-supervised Learning with Generated Data 15**

3.1 Introduction . . . . . 15
3.2 Network Overview . . . . . 18
3.2.1 Generative Adversarial Network . . . . . 19
3.2.2 Convolutional Neural Network . . . . . 20
3.3 The Proposed Regularization Method . . . . . 21
3.3.1 Label Smoothing Regularization Revisit . . . . . 21
3.3.2 Label Smoothing Regularization for Outliers . . . . . 23
3.4 Experiment . . . . . 25
3.4.1 Person Re-id Datasets . . . . . 26
3.4.2 Implementation Details . . . . . 27
3.4.3 Evaluation . . . . . 28
3.4.4 Fine-grained Recognition . . . . . 35
3.5 Summary . . . . . 37

**4 Multi-view Multi-source Image Matching 38**

4.1 Introduction . . . . . 38
4.2 Geo-localization Dataset Review . . . . . 42
4.3 University-1652 Dataset . . . . . 44
4.3.1 Dataset Description . . . . . 44
4.3.2 Evaluation Protocol . . . . . 46
4.4 Cross-view Image Matching . . . . . 47
4.4.1 Visual Representations . . . . . 47

4.4.2 Network Architecture and Loss Function . . . . . 47
4.5 Experiment . . . . . 50
4.5.1 Implementation Details . . . . . 50
4.5.2 Geo-localization Results . . . . . 50
4.5.3 Ablation Study and Further Discussion . . . . . 55
4.6 Summary . . . . . 60

**5 Two-stage Progressive Learning 61**

5.1 Introduction . . . . . 61
5.2 Dataset Collection and Task Definition . . . . . 64
5.2.1 Dataset Analysis . . . . . 64
5.2.2 Task Definition . . . . . 65
5.3 Methodology . . . . . 67
5.3.1 Model Structure . . . . . 67
5.3.2 Two-stage Progressive Learning . . . . . 69
5.3.3 Post-processing . . . . . 72
5.4 Experiment . . . . . 75
5.4.1 Implementation Details . . . . . 75
5.4.2 Quantitative Results . . . . . 77
5.4.3 Further Evaluations and Discussion . . . . . 79
5.5 Summary . . . . . 85

**6 Joint Discriminative and Generative Learning 86**

6.1 Introduction . . . . . 86
6.2 Methodology . . . . . 90
6.2.1 Generative Module . . . . . 91

6.2.2 Discriminative Module . . . . . 94
6.2.3 Optimization . . . . . 96
6.3 Experiment . . . . . 97
6.3.1 Implementation Details . . . . . 98
6.3.2 Generative Evaluations . . . . . 99
6.3.3 Discriminative Evaluations . . . . . 102
6.4 Summary . . . . . 104

**7 Conclusions and Future Work** **107**
7.1 Summary of Contributions . . . . . 107
7.2 Future Directions . . . . . 108

**Bibliography** **111**

# List of Figures

3.1 The pipeline of the proposed method. There are two components: a generative adversarial model [121] for unsupervised learning and a convolutional neural network for semi-supervised learning. "Real Data" represents the labeled data in the given training set; "Training data" includes both the "Real Data" and the generated unlabeled data. Our target is to learn more discriminative embeddings with the "Training data". . . . . 16

3.2 The image distribution per class in the dataset Market-1501 [206], CUHK03 [76] and DukeMTMC-reID (Duke) [126, 217]. We observe that all these datasets suffer from the limited images per class. Note that there are only a few classes with more than 20 images. . . . . 19

3.3 Examples of GAN images and real images. (a) The top two rows show the pedestrian samples generated by DCGAN [121] trained on the Market-1501 training set [206]. (b) The bottom row shows the real samples in training set. Although the generated images in (a) can be easily recognized as fake images by a human, they still serve as an effective regularizer in our experiment. . . . . 20

3.4 The label distributions of a real image and a GAN-generated image in our system. We use a classical label distribution (Eq. 3.2) for the real image (left). For the generated image (right), we employ the proposed LSRO label distribution (Eq. 3.6), *e.g.*a uniform distribution on every training class because the generated image is assumed to belong to none of the training classes. We employ a cross-entropy loss that combines the two types of label distributions as the optimization objective (Eq. 3.7). . . . . 23

3.5 The newly generated images from a DCGAN model trained on DukeMTMC-reID (Duke) and CUB-200-2011. Through LSRO, they are added to the training sets of DukeMTMC-reID and CUB-200-2011 to regularize the CNN model. . . . . 28

3.6 Sample retrieval results on DukeMTMC-reID using the proposed method. The images in the first column are the query images. The retrieved images are sorted according to the similarity scores from left to right. The correct matches are in the blue rectangles, and the false matching images are in the red rectangles. DukeMTMC-reID is challenging because it contains pedestrians with occlusions and similar appearance. . . . . 34

4.1 It is challenging, even for a human, to associate (a) ground-view images with (b) satellite-view images. In this chapter, we introduce a new dataset based on the third platform, *i.e.*, drone, to provide real-life viewpoints and intend to bridge the visual gap against views. (c) Here we show two real drone-view images collected from public drone flights on Youtube [3, 31]. (d) In practice, we use the synthetic drone-view camera to simulate the real drone flight. It is based on two concerns. First, the collection expense of real drone flight is unaffordable. Second, the synthetic camera has a unique advantage in the manipulative viewpoint. Specifically, the 3D engine in Google Earth is utilized to simulate different viewpoints in the real drone camera. . . . . 39

4.2 (a) The drone flight curve toward the target building. When flying around the building, the synthetic drone-view camera could capture rich information of the target, including scale and viewpoint variants. (b) The ground-view images are collected from street-view cameras to obtain different facets of the building as well. It simulates real-world photos when people walk around the building. . 40

4.3 The basic model architectures for cross-view matching. Since the low-level patterns of different data are different, we apply multi-branch CNN to extract high-level features and then build the relation on the high-level features. (I) Model-I is a two-branch CNN model, which only considers the satellite-view and ground-view image matching; (II) Model-II is a two-branch CNN model, which only considers the satellite-view and drone-view image matching; (III) Model-III is a three-branch CNN model, which fully utilizes the annotated data, and considers the images of all three platforms. There are no "standard" methods to build the relationship between the data of multiple sources. Our baseline model applies the instance loss [215] and we also could adopt other loss terms, *e.g.*, triplet loss [14, 24] and contrastive loss [82, 175, 216]. . . . . 48

4.4 The test accuracy curves when using $n$ training drone-view images per class, $n \in \{1, 3, 9, 27, 54\}$. The two sub-figures are the Rank@1 (%) and AP (%) accuracy curves, respectively. The orange curves are for the drone navigation (Satellite $\rightarrow$ Drone), and the blue curves are for the drone-view target localization (Drone $\rightarrow$ Satellite). 53

4.5 Qualitative image retrieval results. We show the top-3 retrieval results of drone-view target localization (left) and drone navigation (right). The results are sorted from left to right according to their confidence scores. The images in yellow boxes are the true matches, and the images in the blur boxes are the false matches. (Best viewed when zoomed in.) . . . . . 54

4.6 Qualitative image search results using real drone-view query. We evaluate the baseline model on an unseen university. There are two results: (I) In the middle column, we use the real drone-view query to search similar synthetic drone-view images. The result suggests that the synthetic data in University-1652 is close to the real drone-view images; (II) In the right column, we show the retrieval results on satellite-view images. It verifies that the baseline model trained on University-1652 has good generalization ability and works well on the real-world query. . . . . 55

5.1 The motivation of our vehicle re-identification method by leveraging public datasets. The common knowledge of discriminating different vehicles could be transferred to the final model. . . . . 64

5.2 (a) The image distribution per class in the vehicle re-id datasets, *e.g.*, CityFlow [148], VehicleID [86] , CompCar [185] and VeRi-776 [95]. We observe that the two largest datasets, *i.e.*, VehicleID and CompCars, suffer from the limited images per class. (b) Here we also provide the image samples of the four datasets. The four datasets contain different visual biases, such as illumination conditions, collection places and viewpoints. . . . . 67

5.3 Illustration of the model structure. We remove the original classifier of the ImageNet pre-trained model, add a new classifier and replace the average pooling with the adaptive average pooling layer. The adaptive average pooling is to squeeze the output to the pre-defined shape (*i.e.*, $1 \times 1$). . . . . 68

5.4 Geometric Interpretation. Here we give a three-class sample to show our intuition. $W_i$ denotes the class weight of the final linear classifier. In this example, the third class denotes one auxiliary class, which belongs to VehicleNet but the target domain. Therefore, in the Stage-II fine-tuning, we remove the auxiliary classes, including $W_3$. The cross-entropy loss of Stage-I pulls the samples with the same label together (close to either the relative weight $W_1$, $W_2$ or $W_3$). In this way, the positive pair is closer than the negative pair, while the samples are far from the decision boundary. Stage I, therefore, leads to a decent weight initialization to be used in Stage II with a large margin from decision boundary, when we leave out the auxiliary class, *i.e.*, the third class with $W_3$, from VehicleNet. . . 71

5.5 The inference pipeline for AICity Challenge Competition. Given one input image and the corresponding cropped image via MaskRCNN [40], we extract features from the trained models, *i.e.*, 8×SE-ResNeXt101 [48]. We normalize and concatenate the features. Meanwhile, we extract the camera prediction from the camera-aware model, *i.e.*, the fine-tuned DenseNet121 [51]. Then query expansion and camera verification are applied. Finally, we utilize the re-ranking technique [220] to retrieve more positive samples. . . . . 72

5.6 Qualitative image search results using the vehicle query images from the CityFlow dataset. We select the four query images from different viewpoints. The results are sorted from left to right according to the similarity score. The true-matches are in green, when the false-matches are in red. . . . . 81

5.7 Visualization of the activation heatmap in the learned model on VehicleNet. The vehicle images in every subfigure (a)-(c) are from the same vehicle ID. Noted that there do exist strong response values at the regions containing discriminative details, such as headlights and tire types. . . . . 81

5.8 The training losses of the two stages. Due to the large-scale data and classes, the first stage (left) takes more epochs to converge. Attribute to the trained weight of the first stage, the second stage (right) converge early. . . . . 85

6.1 Examples of generated images on Market-1501 by switching appearance or structure codes. Each row and column corresponds to different appearance and structure. . . . . 87

6.2 A schematic overview of DG-Net. (a) Our discriminative re-id learning module is embedded in the generative module by sharing appearance encoder $E_a$. A dash black line denotes the input image to structure encoder $E_s$ is converted to gray. The red line indicates the generated images are online fed back to $E_a$. Two objectives are enforced in the generative module: (b) self-identity generation by the same input identity and (c) cross-identity generation by different input identities. (d) To better leverage generated data, the re-id learning involves primary feature learning and fine-grained feature mining. . . . . 91

6.3 Comparison of the generated and real images on Market-1501 across the different methods including LSGAN [104], PG$^2$-GAN [102], FD-GAN [33], PN-GAN [118], and our approach. This figure is best viewed when zoom in. Please attention to both foreground and background of the images. . . . . 97

6.4 Comparison of the generated images by our full model, removing online feeding (w/o feed), and further removing identity supervision (w/o id). . . . . 98

6.5 Example of image generation by linear interpolation between two appearance codes. . . . . 99

6.6 Examples of our generated images by swapping appearance or structure codes on the three datasets. All images are sampled from the test sets. . . . . 100

6.7 Comparison of success and failure cases in our image generation. In the failure case, the logo on t-shirt of the original image is missed in the synthetic image. . . . . 102

6.8 Analysis of the re-id learning related hyper-parameters $\alpha$ and $\beta$ to balance primary and fine-grained features in training (left) and testing (right). . . . . 103

# Chapter 1

# Introduction

Visual matching aims to establish image correspondence across viewpoints. Given one query image, the visual matching system is to find the image containing the object of interest from other viewpoints. The visual matching system can be applied to broad commercial applications, such as product retrieval for online shopping [90, 12], vision-based localization for accurate delivery [210, 230], and traffic management for smart city [148, 147], attracting lots of attention from the community. In recent years, the advance of visual matching is mainly due to two factors: 1) the availability of large-scale datasets and 2) the deeply-learned visual representation. Large-scale datasets facilitate the model training from scratch, which meets the demands of the data-hungry deeply-learned approaches [207, 171]. On the other hand, the rapid development of deeply-learned representation extracted by Convolutional Neural Network (CNN) also provides the breakthrough of the visual representation learning [216, 117, 142].

Despite the great success, the visual matching task remains challenging in the sense that images captured by different cameras often contain significant intra-class variants caused by changes in the background, viewpoint, occlusion and object pose, etc. As a result, designing or learning representations that are robust against intra-class variations as much as possible has been one of the major targets in visual matching. Many efforts have been paid to either mining fine-grained visual features [142, 136] or deep metric learning [43, 140]. In this thesis, we take one different view, focusing on the data limitation and robust learning in visual matching. It is

worthy to note that the annotation for visual matching is generally expensive due to the difficulties of drawing bounding boxes and associating two objects from millions of candidate images. In most recent datasets, despite the large image number in total, the number of training images for each object is still limited, compromising the training process of learning common object variance. For instance, there are 17.2 images per categories in Market-1501 [206], 9.6 images in CUHK03 [76] and 23.5 images in DukeMTMC-reID [126, 217] on average.

## 1.1 Motivation

One straightforward method is to let the model potentially "see" the common variants of objects. **Human could imagine something that did not existed previously and learn from that imagination in order to conceive the idea in the real world [28, 57].** This point inspires me of combining generative learning with discriminative learning to learn one robust system for visual matching. In particular, the generative model provides more high-quality training data with diversity, while discriminative learning motivates the matching model to learn the prior knowledge of potential visual variants.

## 1.2 Approach

With the rapid development of generative adversarial network [36], the generated image is easier to access with relatively good quality. One of the main advantages of using the generated data is that we do not need to collect extra data. Since the generative model is trained on the original training dataset, the generated data generally follows the original data distribution. Before we involve the generated data into training, one remaining question is how to assign one proper label for the newly generated data. For instance, the samples generated by GAN [36] usually contain visual elements from different semantic classes, which is hard to assign an

appropriate label. In Chapter 3, we adopt one label smoothing regularization policy, which views all the unlabeled generated data as outliers for existing categories to regularize the training process.

Another widely-adopted method for data augmentation is to borrow the strength of the synthetic data from 3D simulation systems. In Chapter 4, we collect one multi-view multi-source dataset containing both the real-world data and the synthetic data generated by the 3D engine. We propose one simple but effective model to learn the visual representation. The experiment verifies the effectiveness of the synthetic data to learn the viewpoint-invariant feature for real-world applications.

Except for the large-scale generated data, we also can collect more real-world data from the Internet. The main disadvantage is that the web data is usually from various data sources and contains different characteristics compared with the target dataset. Therefore, the primary challenge we face is the gap between the collected Internet data and the original training data. In Chapter 5, we investigate the two-stage progressive learning strategy for the domain adaptation.

In Chapter 6, we propose one unified framework, which enables data generation and discriminative learning in an end-to-end manner. With the generation quality improvement and the controllable image manipulation, we adopt one new strategy of involving one teacher model to predict more accurate pseudo labels for the generated images. The experiment shows the qualitative and quantitative performance improvement on the image generation task and the image retrieval task.

## 1.3 Contributions

The contributions of this thesis on visual matching are as follows,

1. We propose a novel semi-supervised pipeline that integrates GAN-generated images into the CNN learning machine *in vitro* and propose a Label Smooth-

ing Regularization for Outliers (LSRO) method for semi-supervised learning. The integration of unlabeled data regularizes the CNN learning process. We show that the LSRO method is superior to the two available strategies for dealing with unlabeled data and demonstrate that the proposed semi-supervised pipeline has a consistent improvement over the ResNet baseline on three person re-identification datasets and one fine-grained recognition dataset.

2. The first multi-view multi-source dataset for drone-based geo-localization, University-1652, is contributed. We design effective methods that fully exploit the rich information contained in multi-view data. We also evaluate three basic models and three different loss terms, including contrastive loss [82, 175, 216], triplet loss [14, 24], and instance loss [215]. Apart from the extensive evaluation of the baseline method, we also test the learned model on real drone-view images to evaluate the scalability of the learned feature. Our results show that University-1652 helps the model to learn the viewpoint-invariant feature and reaches a step closer to practice.

3. To address the data limitation, we introduce one large-scale dataset, called VehicleNet, to borrow the strength of the public vehicle datasets, which facilitate the learning of robust vehicle features. In the experiment, we verify the feasibility and effectiveness of learning from VehicleNet. To leverage the multi-source vehicle images in VehicleNet, we propose a simple yet effective learning strategy, *i.e.*, the two-stage progressive learning approach. We discuss and analyze the effectiveness of the two-stage progressive learning approach. The proposed method has achieved competitive performance on the CityFlow benchmark as well as two public vehicle re-identification datasets, *i.e.*, VeRi-776 [95] and VehicleID [86].

4. We provide the first framework that is able to end-to-end integrate discrimi-

native and generative learning in a single unified network for visual matching. Extensive qualitative and quantitative experiments show that our image generation compares favorably against the existing ones, and more importantly, our retrieval accuracy consistently outperforms the competing algorithms by large margins on several benchmarks.

## 1.4 Outline

This thesis is organized into the following chapters:

- *Chapter 2* presents the literature review of existing methods for visual matching, summarising discriminative learning approaches for different vision tasks, recent developments in generative learning, and related works on machine learning studied in this thesis.

- *Chapter 3* provides detailed explanations of the proposed semi-supervised learning pipeline with generated data. It shows that the imperfect generated data can help to regularize the model learning and avoid currently available learning frameworks from the over-fitting problem, yielding consistent improvement.

- *Chapter 4* considers the problem of cross-view geo-localization via synthetic data. Besides phone cameras and satellites, we argue that drones could serve as the third platform to deal with the geo-localization problem. To verify the effectiveness of the drone platform, we introduce a new multi-view multi-source benchmark for drone-based geo-localization, named University-1652. University-1652 contains data from three platforms, i.e., synthetic drones, satellites and ground cameras of 1,652 university buildings around the world. The experiments show that University-1652 helps the model to learn the

viewpoint-invariant features and also has good generalization ability in the real-world scenario.

- *Chapter 5* presents a two-stage progressive learning strategy to leverage the real-world data collected from the web. The first stage of our approach is to learn the generic representation for all domains by training with the conventional classification loss. The second stage is to fine-tune the trained model purely based on the target vehicle set, by minimizing the distribution discrepancy between our VehicleNet and any target domain. We discuss our proposed multi-source dataset VehicleNet and evaluate the effectiveness of the two-stage progressive representation learning through extensive experiments.

- *Chapter 6* explains the joint discriminative and generative learning framework with an end-to-end training manner. The proposed model involves a generative module that separately encodes each person into an appearance code and a structure code, and a discriminative module that shares the appearance encoder with the generative module. The proposed joint learning framework renders significant improvement over the baseline without using generated data, leading to the state-of-the-art performance on several benchmark datasets.

- *Chapter 7* provides conclusions and suggests potential areas to be pursued in the future.

# Chapter 2

# Literature Review

## 2.1 Discriminative Learning

### 2.1.1 Person Re-identification

A large family of person re-identification (re-id) research focuses on metric learning loss. Some methods combine identification loss with verification loss [216, 179], others apply triplet loss with hard sample mining [43, 127, 21]. Several recent works employ pedestrian attributes to enforce more supervisions and perform multi-task learning [84, 137, 164]. Alternatives harness pedestrian alignment and part matching to leverage on the human structure prior. One of the common practices is to split input images or feature maps horizontally to take advantage of local spatial cues [189, 77, 142]. In a similar manner, pose estimation is incorporated into learning local features [136, 204, 172, 138, 218]. Apart from the pose, human parsing is used in [60] to enhance spatial matching. In comparison, our DG-Net (in Chapter 6) relies only on simple identification loss for re-id learning and requires no extra auxiliary information such as pose or human parsing for image generation.

Another active research line is to utilize GANs to augment training data. In [217], Zheng *et al.* first introduce to use unconditional GAN to generate images from random vectors (more details are provided in Chapter 3). Huang *et al.* proceed with this direction with WGAN [5] and assign pseudo labels to generated images [55]. Li *et al.* propose to share weights between re-id model and discriminator of GAN [79]. In addition, some recent methods make use of pose estimation to conduct pose-conditioned image generation. A two-stage generation pipeline is developed in [102]

based on pose to refining generated images. Similarly, pose is also used in [33, 87, 118] to generate images of a pedestrian in different poses to make learned features more robust to pose variances. Siarohin *et al.* achieve better pose-conditioned image generation by using a nearest neighbor loss to replace the traditional $\ell_1$ or $\ell_2$ loss [134]. All the methods set image generation and re-id learning as two disjointed steps, while our DG-Net (in Chapter 6) end-to-end integrates the two tasks into a unified network.

Meanwhile, some recent studies also exploit synthetic data for style transfer of pedestrian images to compensate for the disparity between the source and target domains. CycleGAN [228] is applied in [26, 222] to transfer pedestrian image style from one dataset to another. StarGAN [22] is used in [221] to generate pedestrian images with different camera styles. Bak *et al.* [8] employ a game engine to render pedestrians using various illumination conditions. Wei *et al.* [171] take semantic segmentation to extract foreground mask in assisting style transfer. In contrast to the global style transfer, we aim to manipulate appearance and structure details to facilitate more robust re-id learning in Chapter 6.

### 2.1.2 Vehicle Re-identification

Vehicle re-identification (re-id) demands robust and discriminative image representation. The recent progress of vehicle re-identification has been due to two aspects: 1) the availability of the new vehicle datasets [148, 95, 86, 184] and 2) the discriminative vehicle feature from deeply-learned models [97, 17, 62]. Zapletal *et al.* [193] first collect a large-scale dataset with vehicle pairs and extract the color histograms and oriented gradient histograms feature to discriminate different cars. With the recent advance in Convolutional Neural Network (CNN), Liu *et al.* [91] combine the CNN-based feature with the traditional hand-crafted features to obtain the robust feature. Qian *et al.* [117] and Guo *et al.* [38] propose to aggregate

the multi-level feature to enrich the representation. To take full advantages of the fine-grained patterns, Wang *et al.* [169] first explore the vehicle structure and then extract the part-based CNN features according to the location of key points. Besides, Shen *et al.* [131] involve the temporal-spatial information into the model training as well as the inference process. Another line of works regards vehicle re-identification as a metric learning problem, and explore the objective functions to help the representation learning. Triplet loss has been widely studied in person re-id [43, 215, 27], and also has achieved successes in the vehicle re-id [95]. Zhang *et al.* [202] further company the classification loss with triplet loss, which further improves the re-identification ability. Furthermore, Yan *et al.* [184] propose a multi-grain ranking loss to discriminate the appearance-similar cars. Besides, some works also show the attributes, *e.g.*, color, manufactories and wheel patterns, could help the model to learn the discriminative feature [84, 148, 164].

### 2.1.3 Cross-view Geo-localization

Most previous works treat geo-localization as an image retrieval problem. The key of the geo-localization is to learn the view-point invariant representation, which intends to bridge the gap between images of different views. With the development of the deeply-learned model, convolutional neural networks (CNNs) are widely applied to extract the visual features. One line of works focuses on metric learning and builds the shared space for the images collected from different platforms. Workman *et al.* show that the classification CNN pre-trained on the Place dataset [223] can be very discriminative by itself without explicitly fine-tuning [174]. The contrastive loss, pulling the distance between positive pairs, could further improve the geo-localization results [175, 82]. Recently, Liu *et al.* propose Stochastic Attraction and Repulsion Embedding (SARE) loss, minimizing the KL divergence between the learned and the actual distributions [89]. Another line of works focuses on the

spatial misalignment problem in the ground-to-aerial matching. Vo *et al.* evaluate different network structures and propose an orientation regression loss to train an orientation-aware network [159]. Zhai *et al.* utilize the semantic segmentation map to help the semantic alignment [194], and Hu *et al.* insert the NetVLAD layer [4] to extract discriminative features [49]. Further, Liu *et al.* propose a Siamese Network to explicitly involve the spatial cues, *i.e.*, orientation maps, into the training [88]. Similarly, Shi *et al.* propose a spatial-aware layer to further improve the localization performance [133] and Hu *et al.* [50] also show that spatial alignment is of importance to the geo-localization task. In Chapter 4, since each location has a number of training data from different views, we could train a classification CNN as the basic model. When testing, we use the trained model to extract visual features for the query and gallery images. Then we conduct the feature matching for fast geo-localization.

## 2.2 Generative Learning

### 2.2.1 Generative Adversarial Networks

The generative adversarial networks (GANs) learn two sub-networks: a generator and a discriminator. The discriminator reveals whether a sample is generated or real, while the generator produces samples to cheat the discriminator. The GANs are first proposed by Goodfellow *et al.* [36] to generate images and gain insights into neural networks. Then, DCGANs [121] provides some techniques to improve the stability of training. The discriminator of DCGAN can serve as a robust feature extractor. Salimans *et al.* [130] achieve a state-of-art result in semi-supervised classification and improves the visual quality of GANs. InfoGAN [19] learns interpretable representations by introducing latent codes. On the other hand, GANs also demonstrate potential in generating images for specific fields. Pathak *et al.* [113] propose an encoder-decoder method for image inpainting, where GANs are used as

the image generator. Similarly, Yeh *et al.* [188] improve the inpainting performance by introducing two loss types, and Luo *et al.* [100] introduce the adversarial loss to learn the body structure information for human parsing. In [177], 3D object images are generated by a 3D-GAN, while 2D face images can be automatically made up with different fashion styles [56]. In Chapter 3, we do not focus on investigating more sophisticated sample generation methods. Instead, we use a basic GAN model [121] to generate unlabeled samples from the training data and show that these samples help improve discriminative learning.

### 2.2.2 Dataset Augmentation

Many existing works focus on involving more samples to boost the training. One line of works leverages the generative model to synthesize more samples for training. Wu *et al.* [182] and Yue *et al.* [192] propose to transfer the image into different image styles, *e.g.*, weather conditions, and learn the robust feature for semantic segmentation. In a similar spirit, Zheng *et al.* [217, 211] utilize the Generative Adversarial Network (GAN) [36] to obtain lots of pedestrian images, and then involve the generated samples into training as an extra regularization term. Another line of works collects the real-world data from the Internet to augment the original dataset [209]. One of the pioneering works [65] is to collect a large number of images via searching the keywords on the online engine, *i.e.*, Google. After removing the noisy data, the augmented dataset facilitates the model to achieve state-of-the-art performance on several fine-grained datasets, *e.g.*, CUBird [160]. In a similar spirit, Zheng *et al.* [210] exploit noisy photos of university buildings from Google, benefiting model learning. Besides, several works [139, 147, 187] applies the game engine to build 3D models. Sun *et al.* [139] build a large number of 3D person models, and map models to 2D plane for generating more 2D training data. Yao *et al.* [187] and Tang *et al.* [147] manipulate the generation setting and leverage attributes, *e.g.*, color and

pose, to enable multi-task learning on 2D synthetic data. Lin *et al.* [81] also leverage the synthetic data to learn the common knowledge of human structure, improving the model scalability on real data. In contrast with these existing works, we focus on leveraging the public datasets with different data biases to learn the common knowledge given that vehicles share the similar structure in Chapter 5.

## 2.3 Related Works on Machine Learning

### 2.3.1 Semi-supervised Learning

Semi-supervised learning is a sub-class of supervised learning, taking unlabeled data into consideration, especially when the volume of annotated data is small. On the one hand, some research treats unsupervised learning as an auxiliary task to benefit sequential supervised learning. For example, Hinton *et al.* [45] learn a stack of unsupervised restricted Boltzmann machines to pre-train the model. Ranzato *et al.* [122] propose to reconstruct the input at every level of a network to get a compact representation. In [123], the auxiliary task of ladder networks is to denoise representations at every level of the model. Besides, one line of works is based on the memory mechanism, regularizing the supervised learning during training. As one of the early works, Weston *et al.* [173] propose to use an external memory module to store the long-term memory. In this way, the model could reason with the related experience more effectively. Chen *et al.* [20] apply the memory to the semi-supervised learning to learn from the unlabeled data. Since the historical models memorize the experience from the previous training samples, the temporal ensemble [69] could provide stable and relatively accurate predictions of the unlabeled data. Except for [69], there are different kinds of external memory models. Mean Teacher [149] leverages the weight moving average model as the memory model to regularize the training and French *et al.* [32] extend Mean Teacher for visual domain adaptation. Zhang *et al.* [203] propose mutual learning, which learns the knowledge from multiple student

models. Zheng *et al.* [213] take one step further and leverage the auxiliary classifier in vivo to regularize training. On the other hand, several works assign pseudo labels to the unlabeled data for supervised learning. Papandreou *et al.* [111] combine strong and weak labels in CNNs using an expectation-maximization (EM) process for image segmentation. In [70], Lee assigns a "pseudo label" to the unlabeled data in the class with the maximum predicted probability. In [108, 130], the samples produced by the GAN generator are all taken as one class in the discriminator. Departing from previous semi-supervised works, we adopt a different regularization approach by assigning a uniform label distribution to the generated samples in Chapter 3.

### 2.3.2 Transfer Learning

Transfer learning is to propagate the knowledge of the source domain to the target domain [110]. On the one hand, several recent works focus on the alignment between the source domain and the target domain, which intend to minimize the discrepancy of the two domains. One of the pioneering works [46] is to apply the cyclegan [229] to transfer the image style to the target domain, and then train the model on the transferred data. In this way, the model could learn the similar patterns of the target data. Besides the pixel-level alignment, some works [153, 154, 99, 213] focus on aligning the network activation in the middle or high layers of the neural network. The discriminator is deployed to discriminate the learned feature of the source domain from that of the target domain, and the main target is to minimize the feature discrepancy via adversarial learning. On the other hand, some works deploy pseudo label learning, yielding competitive results as well [232, 233, 71]. The main idea is to make the model more confident to the prediction, which minimizes the information entropy. The pseudo label learning usually contains two steps. The first step is to train one model from scratch on the source domain and generate the pseudo

label for the unlabeled data. The second step is to fine-tune the model and make the model adapt to the target data distribution via the pseudo label. Inspired by the existing works, we propose one simple yet effective two-stage progressive learning. We first train the model on the large-scale VehicleNet dataset and then finetune the model on the target dataset in Chapter 5. The proposed method is also close to the traditional pre-training strategy, but the proposed method could converge quickly and yield competitive performance due to the related vehicle knowledge distilled in the model.

# Chapter 3

# Semi-supervised Learning with Generated Data

## 3.1 Introduction

In this chapter, we propose a semi-supervised pipeline that works on the original training set without an additional data collection process. It is challenging in 1) how to obtain more training data only from the training set and 2) how to use the newly generated data. In this work, the generative adversarial network (GAN) is used to generate unlabeled samples. We propose the label smoothing regularization for outliers (LSRO). This method assigns a uniform label distribution to the unlabeled images, which regularizes the supervised model and improves the baseline. We verify the proposed method on a practical problem: person re-identification (re-ID). This task aims to retrieve a query person from other cameras [207]. We adopt the deep convolutional generative adversarial network (DCGAN) for sample generation, and a baseline convolutional neural network (CNN) for representation learning. Experiments show that adding the GAN-generated data effectively improves the discriminative ability of learned CNN embeddings. On three large-scale datasets, Market1501, CUHK03 and DukeMTMC-reID, we obtain +4.37%, +1.6% and +2.46% improvement in rank-1 precision over the baseline CNN, respectively. We additionally apply the proposed method to fine-grained bird recognition and achieve a +0.6% improvement over a strong baseline.

In particular, this chapter addresses three challenges. First, current research in GANs typically considers the quality of the sample generation with and without semi-supervised learning *in vivo* [108, 130, 121, 19, 113, 177]. Yet a scientific problem

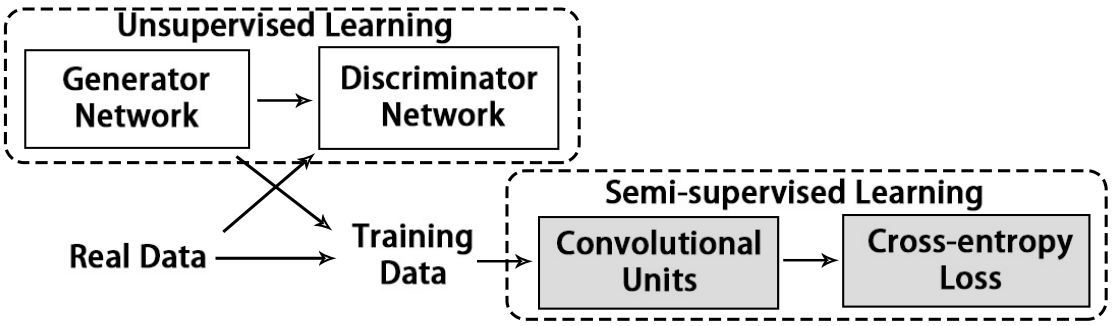


Figure 3.1 : The pipeline of the proposed method. There are two components: a generative adversarial model [121] for unsupervised learning and a convolutional neural network for semi-supervised learning. "Real Data" represents the labeled data in the given training set; "Training data" includes both the "Real Data" and the generated unlabeled data. Our target is to learn more discriminative embeddings with the "Training data".

remains unknown: moving the generated samples out of the box and using them in currently available learning frameworks. To this end, this work uses unlabeled data produced by the DCGAN model [121] in conjunction with the labeled training data. As shown in Figure 3.1, our pipeline feeds the newly generated samples into another learning machine (*i.e.*, a CNN). Therefore, we use the term "*in vitro*" to differentiate our method from [108, 130, 121, 19]; these methods perform semi-supervised learning in the discriminator of the GANs (*in vivo*).

Second, the challenge of performing semi-supervised learning using labeled and unlabeled data in CNN-based methods remains. Usually, the unsupervised data is used as a pre-training step before supervised learning [122, 35, 45]. Our method uses all the data simultaneously. In [111, 70, 108, 130], the unlabeled/weak-labeled real data are assigned labels according to pre-defined training classes, but our method assumes that the GAN generated data does not belong to any of the existing classes.

The proposed LSRO method neither includes unsupervised pre-training nor label assignments for the known classes. We address semi-supervised learning from a new perspective. Since the unlabeled samples do not belong to any of the existing classes, they are assigned a uniform label distribution over the training classes. The network is trained not to predict a particular class for the generated data with high confidence.

Third, in person re-id, data annotation is expensive, because one has to draw a pedestrian bounding box and assign an ID label to it. Recent progress in this field can be attributed to two factors: 1) the availability of large-scale re-id datasets [206, 208, 183, 76] and 2) the learned embedding of pedestrians using a CNN [21, 34]. That being said, the number of images for each identity is still limited, as shown in Figure 3.2. There are 17.2 images per identities in Market-1501 [206], 9.6 images in CUHK03 [76], and 23.5 images in DukeMTMC-reID [126, 217] on average. So using additional data is non-trivial to avoid model overfitting. In the literature, pedestrian images used in training are usually provided by the training sets, without being expanded. So it is unknown if a larger training set with unlabeled images would bring any extra benefit. This observation inspired us to resort to the GAN samples to enlarge and enrich the training set. It also motivated us to employ the proposed regularization to implement a semi-supervised system.

In an attempt to overcome the above-mentioned challenges, this chapter 1) adopts GAN in unlabeled data generation, 2) proposes the label smoothing regularization for outliers (LSRO) for unlabeled data integration, and 3) reports improvements over a CNN baseline on three person re-id datasets. In more details, in the first step, we train DCGAN [121] on the original re-id training set. We generate new pedestrian images by inputting 100-dim random vectors in which each entry falls within [-1, 1]. Some generated samples are shown in Figure 3.3 and Figure 3.5. In the second step, these unlabeled GAN-generated data are fed into the ResNet

model [41]. The LSRO method regularizes the learning process by integrating the unlabeled data and, thus, reduces the risk of over-fitting. Finally, we evaluate the proposed method on person re-id and show that the learned embeddings demonstrate a consistent improvement over the strong ResNet baseline.

To summarize, our contributions are:

- the introduction of a semi-supervised pipeline that integrates GAN-generated images into the CNN learning machine *in vitro*;
- an LSRO method for semi-supervised learning. The integration of unlabeled data regularizes the CNN learning process. We show that the LSRO method is superior to the two available strategies for dealing with unlabeled data; and
- a demonstration that the proposed semi-supervised pipeline has a consistent improvement over the ResNet baseline on three person re-id datasets and one fine-grained recognition dataset.

The main content of this Chapter has been previously published in

*Zhedong Zheng, Liang Zheng, Yi Yang. "Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline in vitro", IEEE International Conference on Computer Vision (ICCV), 2017. (Spotlight)*

## 3.2 Network Overview

In this section, we describe the pipeline of the proposed method. As shown in Figure 3.1, the real data in the training set is used to train the GAN model. Then, the real training data and the newly generated samples are combined into training input for the CNN. In the following section, we will illustrate the structure of the two components, *i.e.*, the GAN and the CNN, in detail. Note that, **our system does not make major changes to the network structures of the**

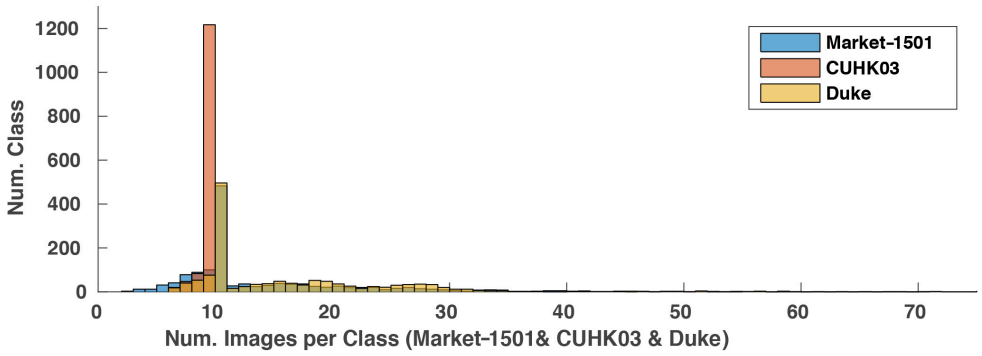


Figure 3.2 : The image distribution per class in the dataset Market-1501 [206], CUHK03 [76] and DukeMTMC-reID (Duke) [126, 217]. We observe that all these datasets suffer from the limited images per class. Note that there are only a few classes with more than 20 images.

**GAN or the CNN with one exception - the number of neurons in the last fully-connected layer in the CNN is modified according to the number of training classes.**

### 3.2.1 Generative Adversarial Network

Generative adversarial networks have two components: a generator and a discriminator. For the generator, we follow the settings in [121]. We start with a 100-dim random vector and enlarge it to $4 \times 4 \times 16$ using a linear function. To enlarge the tensor, five deconvolution functions are used with a kernel size of $5 \times 5$ and a stride of 2. Every deconvolution is followed by a rectified linear unit and batch normalization. Additionally, one optional deconvolutional layer with a kernel size of $5 \times 5$ and a stride of 1, and one *tanh* function are added to fine-tune the result. A sample that is $128 \times 128 \times 3$ in size can then be generated.

The input of the discriminator network includes the generated images and the real images in the training set. We use five convolutional layers to classify whether


Figure 3.3 : Examples of GAN images and real images. (a) The top two rows show the pedestrian samples generated by DCGAN [121] trained on the Market-1501 training set [206]. (b) The bottom row shows the real samples in training set. Although the generated images in (a) can be easily recognized as fake images by a human, they still serve as an effective regularizer in our experiment.

the generated image is fake. Similarly, the size of the convolutional filters is $5 \times 5$ and their stride is 2. We add a fully-connected layer to perform the binary classification (real or fake).

### 3.2.2 Convolutional Neural Network

The ResNet-50 [41] model is used in our experiment. We resize the generated images to $256 \times 256 \times 3$ using bilinear sampling. The generated images are mixed with the original training set as the input of the CNN. That is, the labeled and unlabeled data are simultaneously trained. These training images are shuffled. Following the conventional fine-tuning strategy [207], we use a model pre-trained on ImageNet [129]. We modify the last fully-connected layer to have $K$ neurons to predict the

$K$-classes, where $K$ is the number of the classes in the original training set (as well as the merged new training set). Unlike [108, 130], we do not view the new samples as an extra class but assign a uniform label distribution over the existing classes. So the last fully-connected layer remains $K$-dimensional. The assigned label distribution of the generated images is discussed in the next section.

## 3.3 The Proposed Regularization Method

In this section, we first revisit the label smoothing regularization (LSR), which is used for fully-supervised learning. We then extend LSR to the scenario of unlabeled learning, yielding the proposed label smoothing regularization for outliers (LSRO) method.

### 3.3.1 Label Smoothing Regularization Revisit

LSR was proposed in the 1980s and recently re-discovered by Szegedy *et al.* [144]. In a nutshell, LSR assigns small values to the non-ground truth classes instead of 0. This strategy discourages the network to be tuned towards the ground truth class and thus reduces the chances of over-fitting. LSR is proposed for use with the cross-entropy loss [144].

Formally, let $k \in \{1, 2, ..., K\}$ be the pre-defined classes of the training data, where $K$ is the number of classes. The cross-entropy loss can be formulated as:

$$l = -\sum_{k=1}^{K} \log\left(p(k)\right) q(k), \tag{3.1}$$

where $p(k) \in [0, 1]$ is the predicted probability of the input belonging to class $k$, and can be outputted by CNN. It is derived from the softmax function which normalizes the output of the previous fully-connected layer. $q(k)$ is the ground truth

distribution. Let y be the ground truth class label, $q(k)$ can be defined as:

$$q(k) = \begin{cases} 0 & k \neq y \\ \\ 1 & k = y \end{cases}. \tag{3.2}$$

If we discard the 0 terms in Eq. 3.1, the cross-entropy loss is equivalent to only considering the ground truth term in Eq. 3.3.

$$l = -\log\left(p(y)\right). \tag{3.3}$$

So, minimizing the cross-entropy loss is equivalent to maximizing the predicted probability of the ground-truth class. In [144], the label smoothing regularization (LSR) is introduced to take the distribution of the non-ground truth classes into account. The network is thus encouraged not to be too confident towards the ground truth. In [144], the label distribution $q_{LSR}(k)$ is written as:

$$q_{LSR}(k) = \begin{cases} \frac{\varepsilon}{K} & k \neq y \\ \\ 1 - \varepsilon + \frac{\varepsilon}{K} & k = y \end{cases}, \tag{3.4}$$

where $\varepsilon \in [0, 1]$ is a hyperparameter. If $\varepsilon$ is zero, Eq. 3.4 reduces to Eq. 3.2. If $\varepsilon$ is too large, the model may fail to predict the ground truth label. So in most cases, $\varepsilon$ is set to 0.1. Szegedy *et al.* assume that the non-ground truth classes take on a uniform label distribution. Considering Eq. 3.1 and Eq. 3.4, the cross-entropy loss evolves to:

$$l_{LSR} = -(1 - \varepsilon)\log\left(p(y)\right) - \frac{\varepsilon}{K}\sum_{k=1}^{K}\log\left(p(k)\right). \tag{3.5}$$

Compared with Eq. 3.3, Eq. 3.5 pays additional attention to the other classes, rather than only the ground truth class. In this chapter, we do not employ LSR on the IDE baseline because it yields a slightly lower performance than using Eq. 3.2 (see Section 3.4.3). We re-introduce LSR because it inspires us in designing the LSRO method.


Figure 3.4 : The label distributions of a real image and a GAN-generated image in our system. We use a classical label distribution (Eq. 3.2) for the real image (left). For the generated image (right), we employ the proposed LSRO label distribution (Eq. 3.6), *e.g.*a uniform distribution on every training class because the generated image is assumed to belong to none of the training classes. We employ a cross-entropy loss that combines the two types of label distributions as the optimization objective (Eq. 3.7).

### 3.3.2 Label Smoothing Regularization for Outliers

The label smoothing regularization for outliers (LSRO) is used to incorporate the unlabeled images in the network. This extends LSR from the supervised domain to leverage unsupervised data generated by the GAN.

In LSRO, we propose a virtual label distribution for the unlabeled images. We set the virtual label distribution to be uniform over all classes, due to two inspirations. 1) We assume that the generated samples do not belong to any pre-defined classes. 2) LSR assumes a uniform distribution over the all classes to address overfitting. During testing, we expect that the maximum class probability of a generated

image will be low, *i.e.*, the network will fail to predict a particular class with high confidence. Formally, for a generated image, its class label distribution, $q_{LSRO}(k)$, is defined as:

$$q_{LSRO}(k) = \frac{1}{K}. \tag{3.6}$$

We call Eq. 3.6 the label smoothing regularization for outliers (LSRO).

The one-hot distribution defined in Eq. 3.2 will still be used for the loss computation for the real images in the training set. Combining Eq. 3.2, Eq. 3.6 and Eq. 3.1, we can re-write the cross-entropy loss as:

$$l_{LSRO} = -(1 - Z) \log (p(y)) - \frac{Z}{K} \sum_{k=1}^{K} \log (p(k)). \tag{3.7}$$

For a real training image, $Z = 0$. For a generated training image, $Z = 1$. So our system actually has two types of losses, one for real images and one for generated images.

**Advantage of LSRO.** Using LSRO, we can deal with more training images (outliers) that are located near the real training images in the sample space, and introduce more color, lighting and pose variances to regularize the model. For instance, if we only have one green-clothed identity in the training set, the network may be misled into considering that the color green is a discriminative feature, and this limits the discriminative ability of the model. By adding generated training samples, such as an unlabeled green-clothed person, the classifier will be penalized if it makes the wrong prediction towards the labeled green-clothed person. In this manner, we encourage the network to find more underlying causes and to be less prone to over-fitting. We only use the GAN trained on the original training set to produce outlier images. It would be interesting to further evaluate whether real-world unlabeled images are able to achieve a similar effect (see Table 3.4).

**Competing methods.** We compare LSRO with two alternative methods. Details of both methods are available in existing literature [108, 130, 70]; breif descrip-

tions follow.

- **All in one.** Using [108, 130], a new class label is created, *i.e.*, $K + 1$, and every generated sample is assigned to this class. CNN training follows in Section 3.4.2.

- **Pseudo label.** Using [70], during network training, each incoming GAN-image is passed forward through the current network and is assigned a pseudo label by taking the maximum value of the probability prediction vector ($p(k)$ in Eq. 3.1). This GAN-image can be thus trained in the network with this pseudo label. During training, the pseudo label is assigned *dynamically*, so that the same GAN-image may receive different pseudo labels each time it is fed into the network. In our experiments, we begin feeding GAN images and assigning them pseudo labels after 20 epochs. We also set a global weight to the softmax loss of 0.1 to the GAN and 1 to the real images.

Our experimental results show that the two methods also work on the GAN images and that LSRO is superior to "All in one" and "Pseudo label". Explanations are provided in the Section 3.4.3.

## 3.4 Experiment

We mainly evaluate the proposed method using the Market-1501 [206] dataset, because it is a large scale and has a fixed training/testing split. We also report results on the CUHK03 dataset [76], **but due to the computational cost of 20 training/testing splits, we only use the GAN images generated from the Market-1501 dataset**. In addition, we evaluate our method on a recently released pedestrian dataset DukeMTMC [126] and a fine-grained recognition dataset CUB-200-2011 [160].

### 3.4.1 Person Re-id Datasets

**Market-1501** is a large-scale person re-id dataset collected from six cameras. It contains 19,732 images for testing and 12,936 images for training. The images are automatically detected by the deformable part model (DPM) [30], so the misalignment is common, and the dataset is close to realistic settings. There are 751 identities in the training set and 750 identities in the testing set. There is an average of 17.2 training identity images in the set. We use all the 12,936 detected images from the training set to train the GAN model.

**CUHK03** contains 14,097 images of 1,467 identities. Each identity is captured by two cameras on the CUHK campus. This dataset contains two image sets. One is annotated by hand-drawn bounding boxes, and the other is produced by the DPM detector [30]. We use the detected set in this chapter. There is an average of 9.6 training identity images in the set. We report the averaged result after training/testing 20 times. We use the **single shot** setting.

**DukeMTMC-reID** is a subset of the newly-released multi-target, multi-camera pedestrian tracking dataset [126]. The original dataset contains eight 85-minute high-resolution videos from eight different cameras. Hand-drawn pedestrian bounding boxes are available. In this work, we use a subset of [126] for image-based re-id, in the format of the Market-1501 dataset [206]. We crop pedestrian images from the videos every 120 frames, yielding 36,411 total bounding boxes with IDs annotated by [126]. The DukeMTMC-reID has 1,812 identities from eight cameras. There are 1,404 identities appearing in more than two cameras and 408 identities (distractor ID) who appear in only one camera. We randomly select 702 IDs as the training set and the remaining 702 IDs as the testing set. In the testing set, we pick one query image for each ID in each camera and put the remaining images in the gallery. As a result, we get 16,522 training images with 702 identities, 2,228 query images of the

other 702 identities and 17,661 gallery images. We will release our evaluation protocol. Some example re-id results from the DukeMTMC-reID are shown in Figure 3.6.

### 3.4.2 Implementation Details

**CNN re-id baseline.** We adopt the CNN re-id baseline used in [207, 208]. Specifically, the Matconvnet [158] package is used. During training, We use the ResNet-50 model [41] and modify the fully-connected layer to have 751 and 1,367 neurons for Market-1501 and CUHK03, respectively. All the images are resized to 256 $\times$ 256 before being randomly cropped into 224 $\times$ 224 with random horizontal flipping. We insert a dropout layer before the final convolutional layer and set the dropout rate to 0.5 for CUHK03 and 0.75 for Market-1501 and DukeMTMC-reID, respectively. We use stochastic gradient descent with momentum 0.9. The learning rate of the convolution layers is set to 0.002 and decay to 0.0002 after 40 epochs and we stop training after the 50th epochs. During testing, we extract the 2,048-dim CNN embedding in the last convolutional layer for an input image with a size of 224 $\times$ 224. The similarity between two images is calculated by a cosine distance before ranking.

**GAN training and testing.** We use Tensorflow [1] and the DCGAN package* to train the GAN model using the provided data in the original training set without preprocessing (*e.g.*, foreground detection). All the images are resized to 128 $\times$ 128 and randomly flipped before training. We use Adam [63] with the parameters $\beta_1 = 0.5, \beta_2 = 0.99$. We stop training after 30 epochs. During GAN testing, we input a 100-dim random vector in GAN, and the value of each entry ranges in [-1, 1]. The outputted image is resized to 256 $\times$ 256 and then used in CNN training (with LSRO). More GAN images are shown in Figure 3.5.

*https://github.com/carpedm20/DCGAN-tensorflow


Figure 3.5 : The newly generated images from a DCGAN model trained on DukeMTMC-reID (Duke) and CUB-200-2011. Through LSRO, they are added to the training sets of DukeMTMC-reID and CUB-200-2011 to regularize the CNN model.

### 3.4.3 Evaluation

**The ResNet baseline.** Using the training/testing procedure described in Section 3.4.2, we report the baseline performance of ResNet in Table 3.1, Table 3.5 and Table 3.3. The rank-1 accuracy is 73.69%, 71.5% and 60.28% on Market-1501, CUHK03 and DukeMTMC-reID respectively. Our baseline results are on par with the those reported in [207, 216]. Note that the baseline alone exceeds many previous works [80, 157, 198].

**The GAN images improve the baseline.** As shown in Table 3.2, when we add 24,000 GAN images to the CNN training, our method significantly improves the re-id performance on Market-1501. We observe improvement of +4.37% (from 73.69% to 78.06%) and +4.75% (from 51.48% to 56.23%) in rank-1 accuracy and mAP, respectively. On CUHK03, we observe improvements of +1.6%, +1.2%, +0.8%, and +1.6% in rank-1, 5, 10 accuracy and mAP, respectively. The improve-

ment on CUHK03 is relatively small compared to that of Market-1501, because the DCGAN model is trained on Market-1501 and the generated images share a more similar distribution with Market-1501 than CUHK03. We also observe improvements of +2.46% and +2.14% in rank-1 and mAP, respectively, on the strong ResNet baseline in the DukeMTMC-reID dataset. These results indicate that the unlabeled images generated by the GAN effectively yield improvements over the baseline using the LSRO method.

**The impact of using different numbers of GAN images during training.** We evaluate how the number of GAN images affects the re-id performance. Since the unlabelled data is easy to obtain, we expect the method would learn more general knowledge as the number of unlabelled images increases. The experimental results on Market-1501 are shown in Table 3.2. We note that the number of real training images in Market-1501 is 12,936. Two observations are made.

First, the addition of different numbers of GAN images consistently improves the baseline. Adding approximately 3×GAN images compared to the real training set still has a +2.38% improvement to rank-1 accuracy.

Second, the peak performance is achieved when 2×GAN images are added. When too few GAN sample are incorporated into the system, the regularization ability of the LSRO is inadequate. In contrast, when too many GAN samples are present, the learning machine tends to converge towards assigning uniform prediction probabilities to all the training samples, which is not desirable. Therefore, a trade-off is recommended to avoid poor regularization and over-fitting of uniform label distributions.

**GAN images vs. real images in training.** To further evaluate the proposed method, we replace the GAN images with the real images from the CUHK03 dataset which are viewed as unlabeled in the experiment. Since CUHK03 only 14,097 images,

Table 3.1 : Comparison of the state-of-the-art methods reported on the Market-1501 dataset. We also provide results of the fine-tuned ResNet baseline. Rank-1 precision (%) and mAP (%) are listed. * the respective paper is on ArXiv but not published.

| method | Single Query | | Multi. Query | |
|---|---|---|---|---|
| | rank-1 | mAP | rank-1 | mAP |
| BoW+kissme [206] | 44.42 | 20.76 | - | - |
| BoW+kissme+BQE [85] | 42.55 | 22.39 | - | - |
| MR CNN [155] | 45.58 | 26.11 | 56.59 | 32.26 |
| FisherNet [178] | 48.15 | 29.94 | - | - |
| SL [16] | 51.90 | 26.35 | - | - |
| S-LSTM [157] | - | - | 61.6 | 35.3 |
| DNS [198] | 55.43 | 29.87 | 71.56 | 46.03 |
| Gate Reid [156] | 65.88 | 39.55 | 76.04 | 48.45 |
| SOMAnet [9] | 73.87 | 47.89 | 81.29 | 56.98 |
| Verif.-Identif. [216] | 79.51 | 59.87 | 85.84 | 70.33 |
| DeepTransfer [34]* | 83.7 | 65.5 | **89.6** | 73.8 |
| Basel. [207, 216] | 73.69 | 51.48 | 81.47 | 63.95 |
| Basel. + LSRO | 78.06 | 56.23 | 85.12 | 68.52 |
| Verif-Identif. + LSRO | **83.97** | **66.07** | 88.42 | **76.10** |

Table 3.2 : Comparison of LSRO, "All in one", and "Pseudo label" under different numbers of GAN-generated images on Market-1501. We show that LSRO is superior to the other two methods whose best performance is highlighted in blue and red, respectively. Rank-1 accuracy (%) and mAP (%) are shown.

| # GAN Img. | LSRO | | All in one | | Pseudo label | |
|---|---|---|---|---|---|---|
| | rank-1 | mAP | rank-1 | mAP | rank-1 | mAP |
| 0 (basel.) | 73.69 | 51.48 | 73.69 | 51.48 | 73.69 | 51.48 |
| 12,000 | 76.81 | 55.32 | 75.33 | 52.82 | 76.07 | 53.56 |
| 18,000 | 77.26 | 55.55 | 77.20 | 55.04 | 76.34 | 53.45 |
| 24,000 | **78.06** | **56.23** | 76.63 | 55.12 | 75.80 | 53.03 |
| 30,000 | 77.38 | 55.48 | 75.95 | 55.18 | 75.21 | 52.65 |
| 36,000 | 76.07 | 54.59 | 76.87 | 55.47 | 74.67 | 52.38 |

Table 3.3 : Comparison of the baseline on DukeMTMC-reID. Rank-1 accuracy (%) and mAP (%) are shown.

| method | rank-1 | mAP |
|---|---|---|
| BoW+kissme [206] | 25.13 | 12.17 |
| LOMO+XQDA [80] | 30.75 | 17.04 |
| Basel. [207, 216] | 65.22 | 44.99 |
| Basel. + LSRO | **67.68** | **47.13** |

Table 3.4 : We add the 12,000 real pedestrian images in CUHK03 as outliers to Market-1501. We find the model trained on the generated samples slightly outperforms the model trained on CUHK03 real data. Rank-1 accuracy (%) and mAP (%) are shown.

| Unsup. Data | rank-1 | mAP |
|---|---|---|
| 0 (basel.) | 73.69 | 51.48 |
| CUHK03-Real-12000 | 75.65 | 53.25 |
| Market-1501-GAN-12000 | **76.81** | **55.32** |

we randomly select 12,000 for the fair comparison.

Experimental results are shown in Table 3.4. We compare the results obtained using the 12,000 CUHK03 images and the 12,000 GAN images. We find the real data from CUHK03 also assists in the regularization and improves the performance. But the model trained with GAN-generated data is sightly better. In fact, although the images generated from DCGAN are visually imperfect (see Figure 3.3), they still possess similar regularization ability as the real images.

**Comparison with the two competing methods.** We compare the LSRO method with the "All in one" and "Pseudo label" methods implied in [108, 130] and [70], respectively. The experimental results on Market-1501 are summarized in Table 3.2.

We first observe that both strategies yield improvement over the baseline. The "All in one" method treats all the unlabeled samples as a new class, which forces the network to make "careful" predictions for the existing $K$ classes. The "Pseudo label" method gradually labels the new data, and thus introduces more variance to the network.

Table 3.5 : Comparison of the state-of-the-art reports on the CUHK03 dataset. We list the fine-tuned ResNet baseline as well. The mAP (%) and rank1 (%) precision are presented. * the respective paper is on ArXiv but not published.

| method | rank-1 | rank-5 | rank-10 | mAP |
|---|---|---|---|---|
| KISSME [64] | 11.7 | 33.3 | 48.0 | - |
| DeepReID [76] | 19.9 | 49.3 | 64.7 | - |
| BoW+HS [206] | 24.3 | - | - | - |
| LOMO+XQDA [80] | 46.3 | 78.9 | 88.6 | - |
| SI-CI [163] | 52.2 | 84.3 | 94.8 | - |
| DNS [198] | 54.7 | 80.1 | 88.3 | - |
| SOMAnet [9] | 72.4 | 92.1 | 95.8 | - |
| Verif-Identif. [216] | 83.4 | 97.1 | 98.7 | 86.4 |
| DeepTransfer [34]* | 84.1 | - | - | - |
| Basel. [207, 216] | 71.5 | 91.5 | 95.9 | 75.8 |
| Basel.+LSRO | 73.1 | 92.7 | 96.7 | 77.4 |
| Verif-Identif. + LSRO | **84.6** | **97.6** | **98.9** | **87.4** |


Figure 3.6 : Sample retrieval results on DukeMTMC-reID using the proposed method. The images in the first column are the query images. The retrieved images are sorted according to the similarity scores from left to right. The correct matches are in the blue rectangles, and the false matching images are in the red rectangles. DukeMTMC-reID is challenging because it contains pedestrians with occlusions and similar appearance.

Nevertheless, we find that LSRO exceeds both strategies by approximately +1% $\sim$ +2%. We speculate the reason is that the "All in one" method makes a coarse label estimation, while the "Pseudo label" originally assumes that all the unlabeled data belongs to the existing classes [70] which is not true in person re-id. While these two methods still use the one-hot label distribution, the LSRO method makes a less stronger assumption (label smoothing) towards the labels of the GAN images. These reasons may explain why LSRO has a superior performance.

**Comparison with the state-of-the-art methods.** We compare our method with the state-of-the-art methods on Market-1501 and CUHK03, listed in Table 3.1 and Table 3.5, respectively. On the Market-1501, we achieve **rank-1 accuracy = 78.06%, mAP = 56.23%** when using the single query mode, which is the best result compared to the published papers, and the second best among all the available results including ArXiv papers. On the CUHK03, we arrive at **rank-1 accuracy = 73.1%, mAP = 77.4%** which is also very competitive. The previous best result is produced by combining the identification and the verification losses [34, 216]. We further investigate whether the LSRO could work on this two-stream model. We fine-tuned the publicly available model in [216] with LSRO and achieve state-of-the-art accuracy **rank-1 accuracy = 83.97%, mAP = 66.07%** on Market-1501. On CUHK03, we also observe a state-of-the art performance **rank-1 accuracy = 84.6%, mAP = 87.4%**. We, therefore, show that the LSRO method is complementary to previous methods due to the regularization of the GAN data.

### 3.4.4 Fine-grained Recognition

Fine-grained classification also faces the problem of a lack of training data and annotations. To further test the effectiveness of our method, we provide results on the CUB-200-2011 dataset [160]. This dataset contains 200 bird classes with 29.97

Table 3.6 : We show the recognition accuracy (%) on CUB-200-2011. The proposed method has a 0.6% improvement over the competitive baseline. The two-model ensemble shows a competitive result.

| method | model | annotation | top-1 |
|---|---|---|---|
| Zhang *et al.* [199] | AlexNet | 2×part | 76.7 |
| Zhang *et al.* [199] | VGGNet | 2×part | 81.6 |
| Liu *et al.* [92] | ResNet-50 | attribute | 82.9 |
| Wang *et al.* [162] | 3×VGGNet | × | 83.0 |
| Basel. [92] | ResNet-50 | × | 82.6 |
| Basel.+LSRO | ResNet-50 | × | 83.2 |
| Basel.+LSRO | 2×ResNet-50 | × | **84.4** |

training images per class on average. Bounding boxes are used in both training and testing. We do not use part annotations. In our implementation, the ResNet baseline has a recognition accuracy of 82.6%, which is slightly higher than the 82.3% reported in [92]. This is the baseline we will compare our method with.

Using the same pipeline in Figure 3.1, we train DCGAN on the 5,994 training images with the bounding box, and then we combine the real images with the generated images (see Figure 3.5) to train the CNN. During testing, we adopt the standard 10-crop testing [67], which uses 256 × 256 images as input and the averaged prediction as the classification result. As shown in Table 3.6, the strong baseline alone is superior to some recent methods, and the proposed method further yields an improvement of +0.6% (from 82.6% to 83.2%). We also combine the two models generated by our method with a different initialization to form an ensemble. This leads to an **84.4%** recognition accuracy. In [92], Liu *et al.* report an 85.5%

recognition accuracy with a five-model ensemble using parts and a global scene. We do not include this result because extra annotations are used. We focus on the regularization ability of the GAN, but not on producing a state-of-the-art result.

## 3.5 Summary

In this chapter, we propose an "*in vitro*" usage of the GANs for discriminative learning, *i.e.*, person re-identification. Using a baseline DCGAN model [121], we show that the imperfect GAN images effectively demonstrate their regularization ability when trained with a ResNet baseline network. Through the proposed LSRO method, we mix the unlabeled GAN images with the labeled real training images for simultaneous semi-supervised learning. Albeit simple, we demonstrate consistent performance improvement over the re-id and fine-grained recognition baseline systems, which sheds light on the practical use of GAN-generated data.

In the future, we will continue to investigate on whether GAN images of better visual quality yield superior results when integrated into supervised learning. this chapter provides some baseline evaluations using the imperfect GAN images and the future investigation would be intriguing.

# Chapter 4

# Multi-view Multi-source Image Matching

## 4.1 Introduction

In this chapter, we study the multi-view multi-source image matching problem, *i.e.*, cross-view geo-localization task. Most previous works regard the cross-view geo-localization problem as a sub-task of image retrieval [114, 152, 4, 88, 150, 82, 186, 180]. Given one query image taken at one view, the system aims at finding the most relevant images in another view among large-scale candidates (gallery). Since candidates in the gallery, especially aerial-view images, are annotated with the geographical tag, we can predict the localization of the target place according to the geo-tag of retrieval results. The opportunity for cross-view geo-localization is immense, which could enable subsequent tasks, such as, agriculture, aerial photography, navigation, event detection and accurate delivery [230, 10, 190].

In general, the key to cross-view geo-localization is to learn a discriminative image representation, which is invariant to visual appearance changes caused by viewpoints. Currently, most existing datasets usually provide image pairs and focus on matching the images from two different platforms, *e.g.*, phone cameras and satellites [194, 88]. As shown in Figure 4.1 (a) and (b), the large visual difference between the two images, *i.e.*, ground-view image and satellite-view image, is challenging to matching even for a human. The limited two viewpoints in the training set may also compromise the model to learn the viewpoint-invariant feature.

In light of the above discussions, it is of importance to (1) introduce a multi-view dataset to learn the viewpoint-invariant feature and bridge the visual appearance

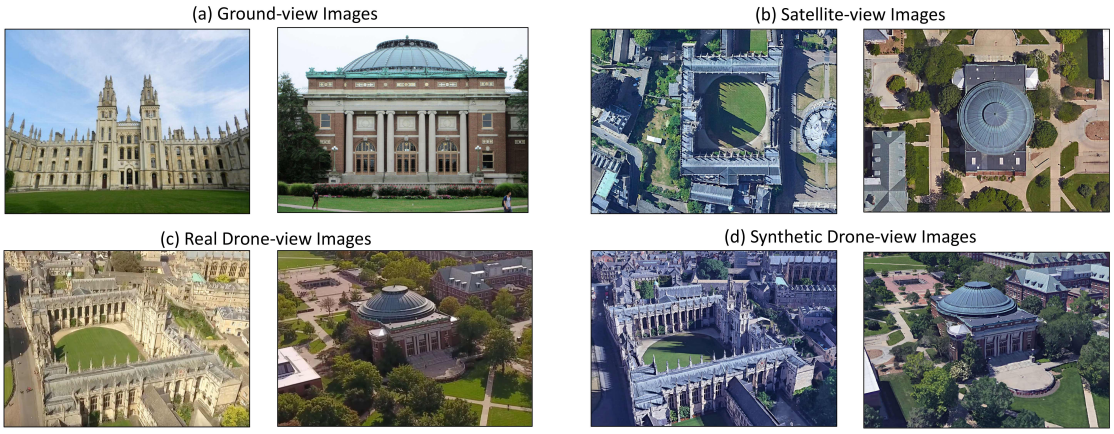


Figure 4.1 : It is challenging, even for a human, to associate (a) ground-view images with (b) satellite-view images. In this chapter, we introduce a new dataset based on the third platform, *i.e.*, drone, to provide real-life viewpoints and intend to bridge the visual gap against views. (c) Here we show two real drone-view images collected from public drone flights on Youtube [3, 31]. (d) In practice, we use the synthetic drone-view camera to simulate the real drone flight. It is based on two concerns. First, the collection expense of real drone flight is unaffordable. Second, the synthetic camera has a unique advantage in the manipulative viewpoint. Specifically, the 3D engine in Google Earth is utilized to simulate different viewpoints in the real drone camera.

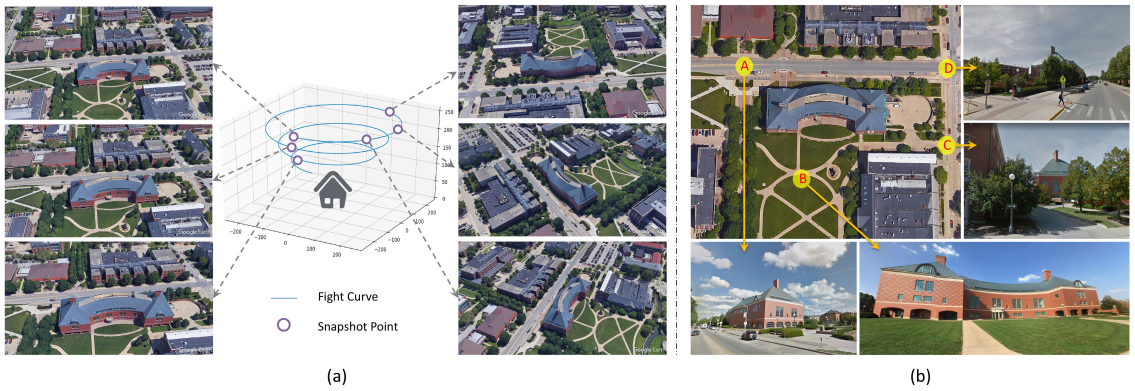


Figure 4.2 : (a) The drone flight curve toward the target building. When flying around the building, the synthetic drone-view camera could capture rich information of the target, including scale and viewpoint variants. (b) The ground-view images are collected from street-view cameras to obtain different facets of the building as well. It simulates real-world photos when people walk around the building.

gap, and (2) design effective methods that fully exploit the rich information contained in multi-view data. With the recent development of the drone [230, 47, 75], we reveal that the drone could serve as a primary data collection platform for cross-view geo-localization (see Figure 4.1 (c) and (d)). Intuitively, drone-view data is more favorable because drones could be motivated to capture rich information of the target place. When flying around the target place, the drone could provide comprehensive views with few obstacles. In contrast, the conventional ground-view images, including *panorama*, inevitably may face occlusions, *e.g.*, trees and surrounding buildings.

However, large-scale real drone-view images are hard to collect due to the high cost and privacy concerns. In light of the recent practice using synthetic training data [125, 87, 181, 73], we propose a multi-view multi-source dataset called University-1652, containing synthetic drone-view images. University-1652 is featured in several aspects. First, it contains multi-view images for every target place. We manipulate the drone-view engine to simulate images of different viewpoints

around the target, which results in 54 drone-view images for every place in our dataset. Second, it contains data from multiple sources. Besides drone-view images, we also collect satellite-view images and ground-view images as reference. Third, it is large-scale, containing $50,218$ training images in total, and has 71.64 images per class on average. The images in the benchmark are captured over $1,652$ buildings of 72 universities. More detailed descriptions will be given in Section 4.3. Finally, University-1652 enables two new tasks, *i.e.*, drone-view target localization and drone navigation.

**Task 1: Drone-view target localization. (Drone $\rightarrow$ Satellite)** Given one drone-view image or video, the task aims to find the most similar satellite-view image to localize the target building in the satellite view.

**Task 2: Drone navigation. (Satellite $\rightarrow$ Drone)** Given one satellite-view image, the drone intends to find the most relevant place (drone-view images) that it has passed by. According to its flight history, the drone could be navigated back to the target place.

In the experiment, we regard the two tasks as cross-view image retrieval problems and compare the generic feature trained on extremely large datasets with the viewpoint-invariant feature learned on the proposed dataset. We also evaluate three basic models and three different loss terms, including contrastive loss [82, 175, 216], triplet loss [14, 24], and instance loss [215]. Apart from the extensive evaluation of the baseline method, we also test the learned model on real drone-view images to evaluate the scalability of the learned feature. Our results show that University-1652 helps the model to learn the viewpoint-invariant feature, and reaches a step closer to practice. Finally, the University-1652 dataset, as well as code for baseline benchmark, will be made publicly available for fair use.

The main content of this Chapter has been previously published in

*Zhedong Zheng, Yunchao Wei, Yi Yang. "University-1652: A Multi-view Multi-source Benchmark for Drone-based Geo-localization", ACM Multimedia (ACM MM), 2020.*

## 4.2 Geo-localization Dataset Review

Most previous geo-localization datasets are based on image pairs, and target matching the images from two different platforms, such as phone cameras and satellites. One of the earliest works [82] proposes to leverage the public sources to build image pairs for the ground-view and aerial-view images. It consists of 78k image pairs from two views, *i.e.*, 45° bird view and ground view. Later, in a similar spirit, Tian *et al.* [150] collect image pairs for urban localization. Differently, they argue that the buildings could serve as an important role to urban localization problem, so they involve building detection into the whole localization pipeline. Besides, the two recent datasets, *i.e.*, CVUSA [194] and CVACT [88], study the problem of matching the panoramic ground-view image and satellite-view image. It could conduct user localization when Global Positioning System (GPS) is unavailable. The main difference between the former two datasets [82, 150] and the later two datasets [194, 88] is that the later two datasets focus on localizing the user, who takes the photo. In contrast, the former two datasets and our proposed dataset focus on localizing the target in the photo. Multiple views towards the target, therefore, are more favorable, which could drive the model to understand the structure of the target as well as help ease the matching difficulty. The existing datasets, however, usually provide the two views of the target place. Different from the existing datasets, the proposed dataset, University-1652, involves more views of the target to boost the viewpoint-invariant feature learning.

Table 4.1 : Comparison between University-1652 and other geo-localization datasets. The existing datasets usually consider matching the images from two platforms, and provide image pairs. In contrast, our dataset focuses on multi-view images, providing 71.64 images per location. For each benchmark, the table shows the number of training images and average images per location, as well as the availability of collection platform, geo-tag, and evaluation metric.

| Datasets | University-1652 | CVUSA [194] | CVACT [88] | Lin *et al.* [82] | Tian *et al.* [150] | Vo *et al.* [159] |
|---|---|---|---|---|---|---|
| #training | 701 × 71.64 | 35.5k × 2 | 35.5k × 2 | 37.5k × 2 | 15.7k × 2 | 900k × 2 |
| Platform | Drone, Ground, Satellite | Ground, Satellite | Ground, Satellite | Ground, 45° Aerial | Ground, 45° Aerial | Ground, Satellite |
| #imgs./location | 54 + 16.64 + 1 | 1 + 1 | 1+1 | 1+1 | 1+1 | 1+1 |
| Target | Building | User | User | Building | Building | User |
| GeoTag | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Evaluation | Recall@K & AP | Recall@K | Recall@K | PR curves & AP | PR curves & AP | Recall@K |

Table 4.2 : Statistics of University-1652 training and test sets, including the image number and the building number of training set, query set and gallery set. We note that there is no overlap in the 33 universities of the training set and 39 universities of test sets.

| Split | #imgs | #classes | #universities |
|---|---|---|---|
| Training | 50,218 | 701 | 33 |
| $\text{Query}_{drone}$ | 37,855 | 701 | 39 |
| $\text{Query}_{satellite}$ | 701 | 701 | |
| $\text{Query}_{ground}$ | 2,579 | 701 | |
| $\text{Gallery}_{drone}$ | 51,355 | 951 | |
| $\text{Gallery}_{satellite}$ | 951 | 951 | |
| $\text{Gallery}_{ground}$ | 2,921 | 793 | |

## 4.3 University-1652 Dataset

### 4.3.1 Dataset Description

In this chapter, we collect satellite-view images, drone-view images with the simulated drone cameras, and ground-view images for every location. We first select 1,652 architectures of 72 universities around the world as target locations. We do not select landmarks as the target. The two main concerns are: first, the landmarks usually contain discriminative architecture styles, which may introduce some unexpected biases; second, the drone is usually forbidden to fly around landmarks. Based on the two concerns, we select the buildings on the campus as the target, which is closer to the real-world practice.

It is usually challenging to build the relation between images from different sources. Instead of collecting data and then finding the connections between various sources, we start by collecting the metadata. We first obtain the metadata of university buildings from Wikipedia *, including building names and university affiliations. Second, we encode the building name to the accurate geo-location, *i.e.*, latitude and longitude, by Google Map. We filter out the buildings with ambiguous search results, and there are 1,652 buildings left. Thirdly, we project the geo-locations in Google Map to obtain the satellite-view images. For the drone-view images, due to the unaffordable cost of the real-world flight, we leverage the 3D models provided by Google Earth to simulate the real drone camera. The 3D model also provides manipulative viewpoints. To enable the scale changes and obtain comprehensive viewpoints, we set the flight curve as a spiral curve (see Figure 4.2(a)) and record the flight video with 30 frames per second. The camera flies around the target with three rounds. The height gradually decreases from 256 meters to 121.5 meters,

*https://en.wikipedia.org/wiki/Category:Buildings_and_structures_by_university_or_college

which is close to the drone flight height in the real world [128, 10].

For ground-view images, we first collect the data from the street-view images near the target buildings from Google Map. Specifically, we manually collect the images in different aspects of the building (see Figure 4.2(b)). However, some buildings do not have the street-view photos due to the accessibility, *i.e.*, most street-view images are collected from the camera on the top of the car. To tackle this issue, we secondly introduce one extra source, *i.e.*, image search engine. We use the building name as keywords to retrieve the relevant images. However, one unexpected observation is that the retrieved images often contain lots of noise images, including indoor environments and duplicates. So we apply the ResNet-18 model trained on the Place dataset [223] to detect indoor images, and follow the setting in [65] to remove the identical images that belong to two different buildings. In this way, we collect $5,580$ street-view images and $21,099$ common-view images from Google Map and Google Image, respectively. It should be noted that images collected from Google Image only serve as an extra training set but a test set.

Finally, every building has 1 satellite-view image, 1 drone-view video, and 3.38 real street-view images on average. We crop the images from the drone-view video every 15 frames, resulting in 54 drone-view images. Overall, every building has totally 58.38 reference images. Further, if we use the extra Google-retrieved data, we will have 16.64 ground-view images per building for training. Compared with existing datasets (see Table 4.1), we summarize the new features in University-1652 into the following aspects:

**1) Multi-source:** University-1652 contains the data from three different platforms, *i.e.*, satellites, drones and phone cameras. To our knowledge, University-1652 is the first geo-localization dataset, containing drone-view images.

**2) Multi-view:** University-1652 contains the data from different viewpoints. The

ground-view images are collected from different facets of target buildings. Besides, synthetic drone-view images capture the target building from various distances and orientations.

**3) More images per class:** Different from the existing datasets that provide image pairs, University-1652 contains 71.64 images per location on average. During the training, more multi-source & multi-view data could help the model to understand the target structure as well as learn the viewpoint-invariant features. At the testing stage, more query images also enable the multiple-query setting. In the experiment, we show that multiple queries could lead to a more accurate target localization.

### 4.3.2 Evaluation Protocol

The University-1652 has $1,652$ buildings in total. There are $1,402$ buildings containing all three views, *i.e.*, satellite-view, drone-view and ground-view images, and 250 buildings that lack either 3D model or street-view images. We evenly split the $1,402$ buildings into the training and test sets, containing 701 buildings of 33 Universities, 701 buildings of the rest 39 Universities. **We note that there are no overlapping universities in the training and test sets.** The rest 250 buildings are added to the gallery as distractors. More detailed statistics are shown in Table 4.2. Several previous datasets [88, 194, 159] adopt the Recall@K, whose value is 1 if the first matched image has appeared before the $K$-th image. Recall@K is sensitive to the position of the first matched image, and suits for the test set with only one true-matched image in the gallery. In our dataset, however, there are multiple true-matched images of different viewpoints in the gallery. The Recall@K could not reflect the matching result of the rest ground-truth images. We, therefore, also adopt the average precision (AP) in [82, 150]. The average precision (AP) is the area under the PR (Precision-Recall) curve, considering all ground-truth images in the gallery. Besides Recall@K, we calculate the AP and report the mean AP value

of all queries.

## 4.4 Cross-view Image Matching

Cross-view image matching could be formulated as a metric learning problem. The target is to map the images of different sources to a shared space. In this space, the embeddings of the same location should be close, while the embeddings of different locations should be apart.

### 4.4.1 Visual Representations

There are no "standard" visual representations for the multi-source multi-view dataset, which demands robust features with good scalability towards different kinds of input images. In this work, we mainly compare two types of features: (1) the generic deep-learned features trained on extremely large datasets, such as ImageNet [25], Place-365 [223], and SfM-120k [120]; (2) the learned feature on our dataset. For a fair comparison, the backbone of all networks is ResNet-50 [41] if not specified. More details are in Section 4.5.2. Next, we describe the learning method on our data in the following section.

### 4.4.2 Network Architecture and Loss Function

The images from different sources may have different low-level patterns, so we denote three different functions $\mathcal{F}_s$, $\mathcal{F}_g$, and $\mathcal{F}_d$, which project the input images from satellites, ground cameras and drones to the high-level features. Specifically, to learn the projection functions, we follow the common practice in [82, 88], and adopt the two-branch CNN as one of our basic structures. To verify the priority of the drone-view images to the ground-view images, we introduce two basic models for different inputs (see Figure 4.3 (I),(II)). Since our dataset contains data from three different sources, we also extend the basic model to the three-branch CNN to fully leverage the annotated data (see Figure 4.3 (III)).

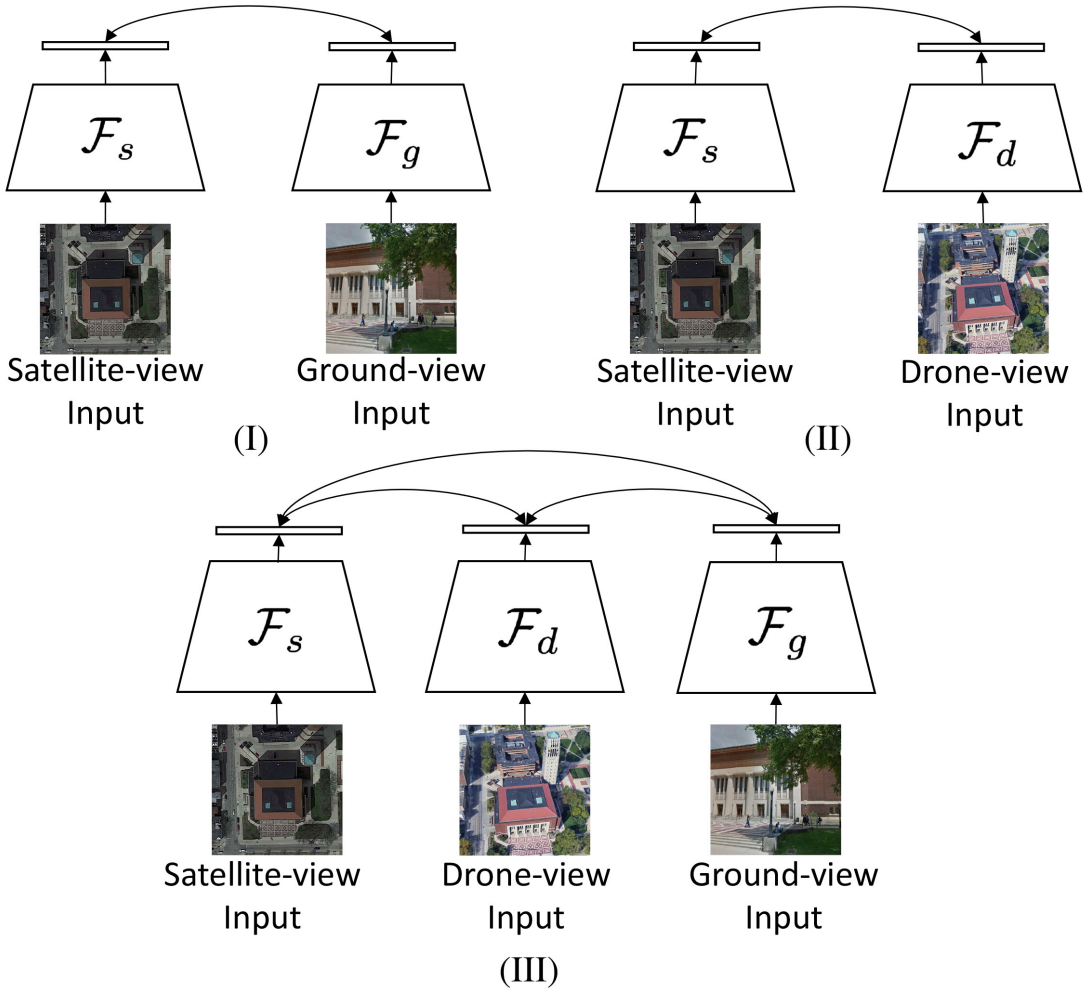


Figure 4.3 : The basic model architectures for cross-view matching. Since the low-level patterns of different data are different, we apply multi-branch CNN to extract high-level features and then build the relation on the high-level features. (I) Model-I is a two-branch CNN model, which only considers the satellite-view and ground-view image matching; (II) Model-II is a two-branch CNN model, which only considers the satellite-view and drone-view image matching; (III) Model-III is a three-branch CNN model, which fully utilizes the annotated data, and considers the images of all three platforms. There are no "standard" methods to build the relationship between the data of multiple sources. Our baseline model applies the instance loss [215] and we also could adopt other loss terms, *e.g.*, triplet loss [14, 24] and contrastive loss [82, 175, 216].

To learn the semantic relationship, we need one objective to bridge the gap between different views. Since our datasets provide multiple images for every target place, we could view every place as one class to train a classification model. In light of the recent development in image-language bi-directional retrieval, we adopt one classification loss called instance loss [215] to train the baseline. The main idea is that a shared classifier could enforce the images of different sources mapping to one shared feature space. We denote $x_s$, $x_d$, and $x_g$ as three images of the location $c$, where $x_s$, $x_d$, and $x_g$ are the satellite-view image, drone-view image and ground-view image, respectively. Given the image pair $\{x_s, x_d\}$ from two views, the basic instance loss could be formulated as:

$$p_s = softmax(W_{share} \times \mathcal{F}_s(x_s)), \tag{4.1}$$

$$L_s = -\log(p_s(c)), \tag{4.2}$$

$$p_d = softmax(W_{share} \times \mathcal{F}_d(x_d)), \tag{4.3}$$

$$L_d = -\log(p_d(c)), \tag{4.4}$$

where $W_{share}$ is the weight of the last classification layer. $p(c)$ is the predicted possibility of the right class $c$. Different from the conventional classification loss, the shared weight $W_{share}$ provides a soft constraint on the high-level features. We could view the $W_{share}$ as one linear classifier. After optimization, different feature spaces are aligned with the classification space. In this chapter, we further extend the basic instance loss to tackle the data from multiple sources. For example, if one more view is provided, we only need to include one more criterion term:

$$p_g = softmax(W_{share} \times \mathcal{F}_g(x_g)), \tag{4.5}$$

$$L_g = -\log(p_g(c)), \tag{4.6}$$

$$L_{total} = L_s + L_d + L_g. \tag{4.7}$$

Note that we keep $W_{share}$ for the data from extra sources. In this way, the soft constraint also works on extra data. In the experiment, we show that the instance loss

objective $L_{total}$ works effectively on the proposed University-1652 dataset. We also compare the instance loss with the widely-used triplet loss [14, 24] and contrastive loss [82, 175, 216] with hard mining policy [43, 109] in Section 4.5.3.

## 4.5 Experiment

### 4.5.1 Implementation Details

We adopt the ResNet-50 [41] pretrained on ImageNet [25] as our backbone model. We remove the original classifier for ImageNet and insert one 512-dim fully-connected layer and one classification layer after the pooling layer. The model is trained by stochastic gradient descent with momentum 0.9. The learning rate is 0.01 for the new-added layers and 0.001 for the rest layers. Dropout rate is 0.75. While training, images are resized to $256 \times 256$ pixels. We perform simple data augmentation, such as horizontal flipping. For satellite-view images, we also conduct random rotation. When testing, we use the trained CNN to extract the corresponding features for different sources. The cosine distance is used to calculate the similarity between the query and candidate images in the gallery. The final retrieval result is based on the similarity ranking. If not specified, we deploy the Model-III, which fully utilizes the annotated data as the baseline model. We also share the weights of $\mathcal{F}_s$ and $\mathcal{F}_d$, since the two sources from aerial views share some similar patterns.

### 4.5.2 Geo-localization Results

To evaluate multiple geo-localization settings, we provide query images from source $A$ and retrieve the relevant images in gallery $B$. We denote the test setting as $A \rightarrow B$.

**Generic features vs. learned features.** We evaluate two categories of features: 1) the generic CNN features. Some previous works [175] show that the CNN model trained on either ImageNet [25] or PlaceNet [223] has learned discriminative feature

Table 4.3 : Comparison between generic CNN features and the learned feature on the University-1652 dataset. The learned feature is shorter than the generic features but yields better accuracy. R@K (%) is Recall@K, and AP (%) is average precision (high is good).

| Training Set | Feature | Drone → Satellite | | Satellite → Drone | |
|---|---|---|---|---|---|
| | Dim | R@1 | AP | R@1 | AP |
| ImageNet [25] | 2048 | 10.11 | 13.04 | 33.24 | 11.59 |
| Place365 [223] | 2048 | 5.21 | 6.98 | 20.40 | 5.42 |
| SfM-120k [120] | 2048 | 12.53 | 16.08 | 37.09 | 10.28 |
| University-1652 | 512 | 58.49 | 63.13 | 71.18 | 58.74 |

Table 4.4 : Ground-view query vs. drone-view query. $m$ denotes multiple-query setting. The result suggests that drone-view images are superior to ground-view images when retrieving satellite-view images.

| Query → Gallery | R@1 | R@5 | R@10 | AP |
|---|---|---|---|---|
| Ground → Satellite | 1.20 | 4.61 | 7.56 | 2.52 |
| Drone → Satellite | 58.49 | 78.67 | 85.23 | 63.13 |
| $m$Ground → Satellite | 1.71 | 6.56 | 10.98 | 3.33 |
| $m$Drone → Satellite | 69.33 | 86.73 | 91.16 | 73.14 |

by itself. We extract the feature before the final classification layer. The feature dimension is 2048. Besides, we also test the widely-used place recognition model [120], whose backbone is ResNet-101. 2) the CNN features learned on our dataset. Since we add one fully-connected layer before the classification layer, our final feature is 512-dim. As shown in Table 4.3, our basic model achieves much better performance with the shorter feature length, which verifies the effectiveness of the proposed baseline.

**Ground-view query vs. drone-view query.** We argue that drone-view images are more favorable comparing to ground-view images, since drone-view images are taken from a similar viewpoint, *i.e.*, aerial view, with the satellite images. Meanwhile, drone-view images could avoid obstacles, *e.g.*, trees, which is common in the ground-view images. To verify this assumption, we train the baseline model and extract the visual features of three kinds of data. As shown in Table 4.4, when searching the relevant satellite-view images, the drone-view query is superior to the ground-view query. Our baseline model using drone-view query has achieved 58.49% Rank@1 and 63.13% AP accuracy.

**Multiple queries.** Further, in the real-world scenario, one single image could not provide a comprehensive description of the target building. The user may use multiple photos of the target building from different viewpoints as the query. For instance, we could manipulate the drone fly around the target place to capture multiple photos. We evaluate the multiple-query setting by directly averaging the query features [206]. Searching with multiple drone-view queries generally arrives higher accuracy with about 10% improvement in Rank@1 and AP, comparing with the single-query setting (see Table 4.4). Besides, the target localization using the drone-view queries still achieves better performance than ground-view queries by a large margin. We speculate that the ground-view query does not work well in the single-query setting, which also limits the performance improvement in the multiple-

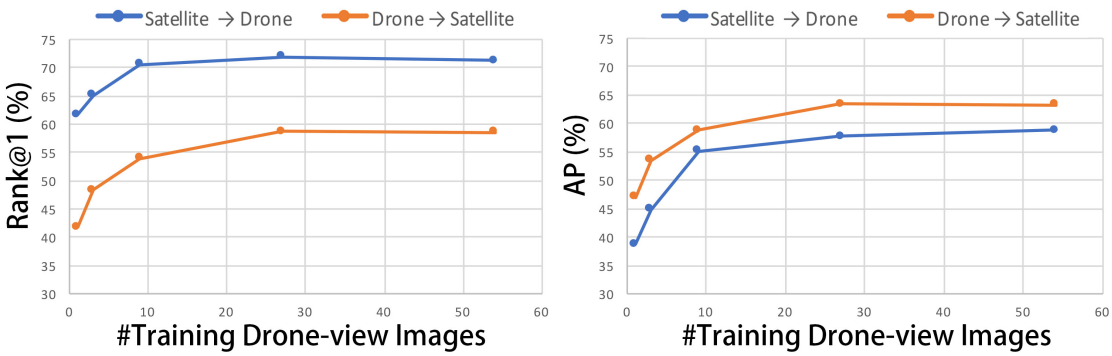


Figure 4.4 : The test accuracy curves when using $n$ training drone-view images per class, $n \in \{1, 3, 9, 27, 54\}$. The two sub-figures are the Rank@1 (%) and AP (%) accuracy curves, respectively. The orange curves are for the drone navigation (Satellite → Drone), and the blue curves are for the drone-view target localization (Drone → Satellite).

query setting.

**Does multi-view data help the viewpoint-invariant feature learning?** Yes. We fix the hyper-parameters and only modify the number of drone-view images in the training set. We train five models with $n$ drone-view images per class, where $n \in \{1, 3, 9, 27, 54\}$. As shown in Figure 4.4, when we gradually involve more drone-view training images from different viewpoints, the Rank@1 accuracy and AP accuracy both increase.

**Does the learned model work on the real data?** Yes. Due to the cost of collecting real drone-view videos, here we provide a qualitative experiment. We collect one 4K real drone-view video of University-X from Youtube granted by the author. University-X is one of the schools in the test set, and the baseline model has not seen any samples from University-X. We crop images from the video to evaluate the model. In Figure 4.6, we show the two retrieval results, *i.e.*, Real Drone → Synthetic Drone, Real Drone → Satellite. The first retrieval result is to verify

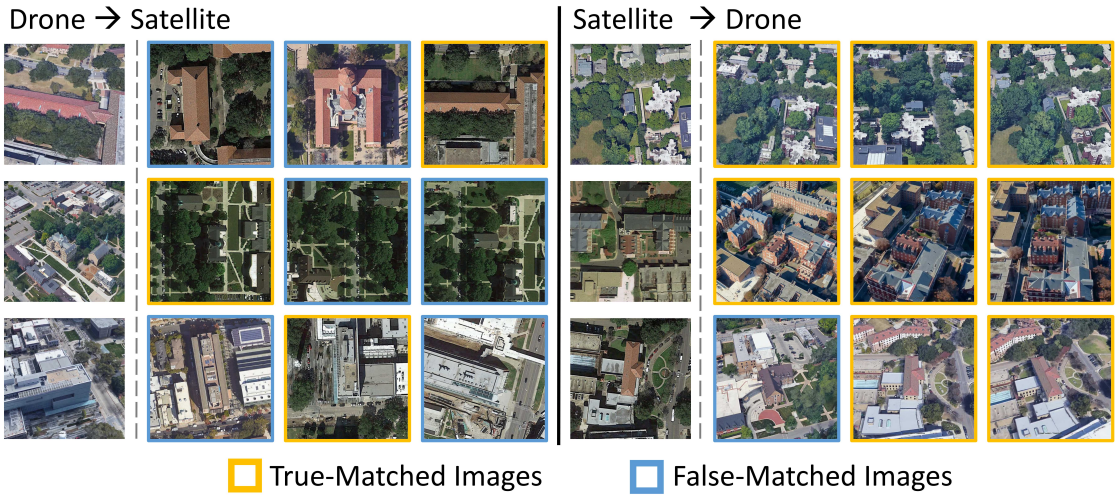


Figure 4.5 : Qualitative image retrieval results. We show the top-3 retrieval results of drone-view target localization (left) and drone navigation (right). The results are sorted from left to right according to their confidence scores. The images in yellow boxes are the true matches, and the images in the blur boxes are the false matches. (Best viewed when zoomed in.)

whether our synthetic data well simulates the images in the real drone cameras. We show the top-5 similar images in the test set retrieved by our baseline model. It demonstrates that the visual feature of the real drone-view query is close to the feature of our synthetic drone-view images. The second result on the Real Drone → Satellite is to verify the generalization of our trained model on the real drone-view data. We observe that the baseline model has good generalization ability and also works on the real drone-view images for drone-view target localization. The true-matched satellite-view images are all retrieved in the top-5 of the ranking list.

**Visualization.** For additional qualitative evaluation, we show retrieval results by our baseline model on University-1652 test set (see Figure 4.5). We can see that the baseline model is able to find the relevant images from different viewpoints. For the false-matched images, although they are mismatched, they share some similar

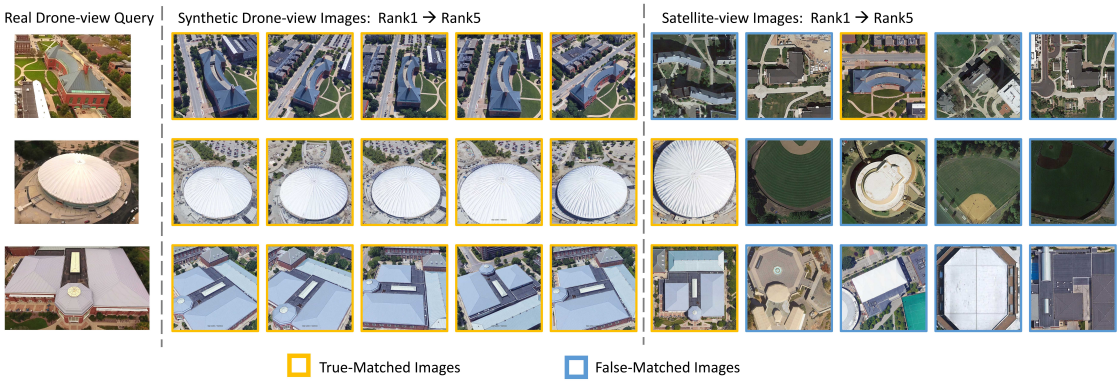


Figure 4.6 : Qualitative image search results using real drone-view query. We evaluate the baseline model on an unseen university. There are two results: (I) In the middle column, we use the real drone-view query to search similar synthetic drone-view images. The result suggests that the synthetic data in University-1652 is close to the real drone-view images; (II) In the right column, we show the retrieval results on satellite-view images. It verifies that the baseline model trained on University-1652 has good generalization ability and works well on the real-world query.

structure pattern with the query image.

### 4.5.3 Ablation Study and Further Discussion

**Effect of loss objectives.** The triplet loss and contrastive loss are widely applied in previous works [82, 175, 24, 14, 216], and the weighted soft margin triplet loss is deployed in [49, 88, 11]. We evaluate these three losses on two tasks, *i.e.*, Drone → Satellite and Satellite → Drone and compare three losses with the instance loss used in our baseline. For a fair comparison, all losses are trained with the same backbone model and only use drone-view and satellite-view data as the training set. For the triplet loss, we also try two common margin values $\{0.3, 0.5\}$. In addition, the hard sampling policy is also applied to these baseline methods during training [43, 109]. As shown in Table 4.5, we observe that the model with instance loss arrives better

Table 4.5 : Ablation study of different loss terms. To fairly compare the five loss terms, we trained the five models on satellite-view and drone-view data, and hold out the ground-view data. For contrastive loss, triplet loss and weighted soft margin triplet loss, we also apply the hard-negative sampling policy.

| Loss | Drone → Satellite | | Satellite → Drone | |
|---|---|---|---|---|
| | R@1 | AP | R@1 | AP |
| Contrastive Loss | 52.39 | 57.44 | 63.91 | 52.24 |
| Triplet Loss (margin=0.3) | 55.18 | 59.97 | 63.62 | 53.85 |
| Triplet Loss (margin=0.5) | 53.58 | 58.60 | 64.48 | 53.15 |
| Weighted Soft Margin Triplet Loss | 53.21 | 58.03 | 65.62 | 54.47 |
| Instance Loss | 58.23 | 62.91 | 74.47 | 59.45 |

Table 4.6 : Ablation study. With/without sharing CNN weights on University-1652. The result suggests that sharing weights could help to regularize the CNN model.

| Method | Drone → Satellite | | Satellite → Drone | |
|---|---|---|---|---|
| | R@1 | AP | R@1 | AP |
| Not sharing weights | 39.84 | 45.91 | 50.36 | 40.71 |
| Sharing weights | 58.49 | 63.31 | 71.18 | 58.74 |

Table 4.7 : Ablation study of different input sizes on the University-1652 dataset.

| Image Size | Drone → Satellite | | Satellite → Drone | |
|---|---|---|---|---|
| | R@1 | AP | R@1 | AP |
| 256 | 58.49 | 63.31 | 71.18 | 58.74 |
| 384 | 62.99 | 67.69 | 75.75 | 62.09 |
| 512 | 59.69 | 64.80 | 73.18 | 59.40 |

Table 4.8 : Comparison of the three CNN models mentioned in Figure 4.3. R@K (%) is Recall@K, and AP (%) is average precision (high is good). Model-III that utilizes all annotated data outperforms the other two models in the three of four tasks.

| Model | Training Set | Drone → Satellite | | | Satellite → Drone | | | Ground → Satellite | | | Satellite → Ground | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | R@1 | R@10 | AP | R@1 | R@10 | AP | R@1 | R@10 | AP | R@1 | R@10 | AP |
| Model-I | Satellite + Ground | - | - | - | - | - | - | 0.62 | 5.51 | 1.60 | 0.86 | 5.99 | 1.00 |
| Model-II | Satellite + Drone | 58.23 | 84.52 | 62.91 | 74.47 | 83.88 | 59.45 | - | - | - | - | - | - |
| Model-III | Satellite + Drone + Ground | 58.49 | 85.23 | 63.13 | 71.18 | 82.31 | 58.74 | 1.20 | 7.56 | 2.52 | 1.14 | 8.56 | 1.41 |

performance than the triplet loss and contrastive loss on both tasks.

**Effect of sharing weights.** In our baseline model, $\mathcal{F}_s$ and $\mathcal{F}_d$ share weights, since two aerial sources have some similar patterns. We also test the model without sharing weights (see Table 4.6). The performance of both tasks drops. The main reason is that limited satellite-view images (one satellite-view image per location) are prone to be overfitted by the separate CNN branch. When sharing weights, drone-view images could help regularize the model, and the model, therefore, achieves better Rank@1 and AP accuracy.

**Effect of the image size.** Satellite-view images contain the fine-grained information, which may be compressed with small training size. We, therefore, try to enlarge the input image size and train the model with the global average pooling. The dimension of the final feature is still 512. As shown in Table 4.7, when we increase the input size to 384, the accuracy of both task, drone-view target localization (Drone → Satellite) and drone navigation (Satellite → Drone) increases. However, when we increase the size to 512, the performance drops. We speculate that the larger input size is too different from the size of the pretrained weight on ImageNet, which is $224 \times 224$. As a result, the input size of 512 does not perform well.

**Different baseline models.** We evaluate three different baseline models as discussed in Section 4.4. As shown in Table 4.8, there are two main observations: 1). Model-II has achieved better Rank@1 and AP accuracy for drone navigation (Satellite $\rightarrow$ Drone). It is not surprising since Model-II is only trained on the drone-view and satellite-view data. 2). Model-III, which fully utilizes all annotated data, has achieved the best performance in the three of all four tasks. It could serve as a strong baseline for multiple tasks.

**Proposed baseline on the other benchmark.** As shown in Table 4.9, we also evaluate the proposed baseline on one widely-used two-view benchmark, *e.g.*, CVUSA [194]. For fair comparison, we also adopt the 16-layer VGG [135] as the backbone model. We do not intend to push the state-of-the-art performance but to show the flexibility of the proposed baseline, which could also work on the conventional dataset. We, therefore, do not conduct tricks, such as image alignment [132] or feature ensemble [124]. Our intuition is to provide one simple and flexible baseline to the community for further evaluation. Compared with the conventional Siamese network with triplet loss, the proposed method could be easily extended to the training data from $N$ different sources ($N \geq 2$). The users only need to modify the number of CNN branches. Albeit simple, the experiment verifies that the proposed method could serve as a strong baseline and has good scalability toward real-world samples.

**Transfer learning from University-1652 to small-scale datasets.** We evaluate the generalization ability of the baseline model on two small-scale datasets, *i.e.*, Oxford [114] and Pairs [115]. Oxford and Pairs are two popular place recognition datasets. We directly evaluate our model on these two datasets without finetuning. Further, we also report results on the revised Oxford and Paris datasets (denoted as ROxf and RPar) [119]. In contrast to the generic feature trained on ImageNet [25], the learned feature on University-1652 shows better generalization ability. Specifi-

Table 4.9 : Comparison of results on the two-view dataset CVUSA [194] with VGG-16 backbone. †: The method utilizes extra orientation information as input.

| Methods | R@1 | R@5 | R@10 | R@Top1% |
|---|---|---|---|---|
| Workman [175] | - | - | - | 34.40 |
| Zhai [194] | - | - | - | 43.20 |
| Vo [159] | - | - | - | 63.70 |
| CVM-Net [49] | 18.80 | 44.42 | 57.47 | 91.54 |
| Orientation [88]† | 27.15 | 54.66 | 67.54 | **93.91** |
| Ours | **43.91** | **66.38** | **74.58** | 91.78 |

Table 4.10 : Transfer learning from University-1652 to small-scale datasets. We show the AP (%) accuracy on Oxford [114], Paris [115], ROxford and RParis [119]. For ROxford and RParis, we report results in both medium (M) and hard (H) settings.

| Method | Oxford | Paris | ROxf (M) | RPar (M) | ROxf (H) | RPar (H) |
|---|---|---|---|---|---|---|
| ImageNet | 3.30 | 6.77 | 4.17 | 8.20 | 2.09 | 4.24 |
| $\mathcal{F}_s$ | 9.24 | 13.74 | 5.83 | 13.79 | 2.08 | 6.40 |
| $\mathcal{F}_g$ | 25.80 | 28.77 | 15.52 | 24.24 | 3.69 | 10.29 |

cally, we try two different branches, *i.e.*, $\mathcal{F}_s$ and $\mathcal{F}_g$, to extract features. $\mathcal{F}_s$ and $\mathcal{F}_g$ share the high-level feature space but pay attention to different low-level patterns of inputs from different platforms. $\mathcal{F}_s$ is learned on satellite-view images and drone-view images, while $\mathcal{F}_g$ learns from ground-view images. As shown in Table 4.10, $\mathcal{F}_g$ has achieved better performance than $\mathcal{F}_s$. We speculate that there are two main reasons. First, the test data in Oxford and Pairs are collected from Flickr, which is closer to the Google Street View images and the images retrieved from Google Image in the ground-view data. Second, $\mathcal{F}_s$ pay more attention to vertical viewpoint changes instead of horizontal viewpoint changes, which are common in Oxford and Paris.

## 4.6 Summary

This chapter contributes a multi-view multi-source benchmark called University-1652. University-1652 contains the data from three platforms, including satellites, drones and ground cameras, and enables the two new tasks, *i.e.*, drone-view target localization and drone navigation. We view the two tasks as the image retrieval problem, and present the baseline model to learn the viewpoint-invariant feature. In the experiment, we observe that the learned baseline model has achieved competitive performance towards the generic feature, and shows the feasibility of drone-view target localization and drone navigation. In the future, we will continue to investigate more effective and efficient feature of the two tasks. One extension of this chapter via mining local viewpoint-invariant patterns has been published on TCSVT 2021 [166].

# Chapter 5

# Two-stage Progressive Learning

## 5.1 Introduction

Vehicle re-identification (re-id) is to spot the car of interest in different cameras and is usually viewed as a sub-task of image retrieval problem [211]. It could be applied to the public place for the traffic analysis, which facilitates the traffic jam management and the flow optimization [148]. Yet vehicle re-id remains challenging since it inherently contains multiple intra-class variants, such as viewpoints, illumination and occlusion. Thus, vehicle re-id system demands a robust and discriminative visual representation given that the realistic scenarios are diverse and complicated. Recent years, Convolutional Neural Network (CNN) has achieved the state-of-the-art performance in many computer vision tasks, including person re-id [142, 141, 218] and vehicle re-id [95, 226, 169], but CNN is data-hungry and prone to over-fitting small-scale datasets. Since the paucity of vehicle training images compromises the learning of robust features, vehicle re-id for the small datasets turn into a challenging problem.

One straightforward approach is to annotate more data and re-train the CNN-based model on the augmented dataset. However, it is usually unaffordable due to the annotation difficulty and the time cost. Considering that many vehicle datasets collected in lab environments are publicly available, an interesting problem arises: Can we leverage the public vehicle image datasets to learn the robust vehicle representation? Given vehicle datasets are related and vehicles share the similar structure, more data from different sources could help the model to learn the common knowl-

edge of vehicles. Inspired by the success of large-scale datasets, *e.g.*, ImageNet [25], we collect a large-scale vehicle dataset, called VehicleNet.

Intuitively, we could utilize VehicleNet to learn the relevance between different vehicle re-id datasets. Then the robust features could be obtained by minimizing the objective function. However, different datasets are collected in different environments, and contains different biases. Some datasets, such as CompCar [185], are mostly collected in the car exhibitions, while other datasets, *e.g.*, City-Flow [148] and VeRi-776 [95], are collected in the real traffic scenarios. Thus, another scientific problem of how to leverage the multi-source vehicle dataset occurs. In several existing works, some researchers resort to transfer learning [110], which aims at transferring the useful knowledge from the labeled source domain to the unlabeled target domain and minimizing the discrepancy between the source domain and the target domain. Inspired by the spirit of transfer learning, in this work, we propose a simple two-stage progressive learning strategy to learn from VehicleNet and adapt the trained model to the realistic environment.

In a summary, to address the above-mentioned challenges, *i.e.*, the data limitation and the usage of multi-source dataset, we propose to build a large-scale dataset, called VehicleNet, via the public datasets and learn the common knowledge of the vehicle representation via two-stage progressive learning (see Figure 5.1). Specifically, instead of only using the original training dataset, we first collect free vehicle images from the web. Comparing with the training set of the CityFlow dataset, we scale up the number of training images from $26,803$ to $434,440$ as a new dataset called VehicleNet. We train the CNN-based model to identify different vehicles, and extract features. With the proposed two-stage progressive learning, the model is further fine-tuned to adapt to the target data distribution, yielding the performance boost. In the experiment, we show that it is feasible to train models with a combination of multiple datasets. When training the model with more samples,

we observe a consistent performance boost, which is consistent with the observation in some recent works [211, 65, 103]. Without explicit vehicle part matching or attribute recognition, the CNN-based model learns the viewpoint-invariant feature by "seeing" more vehicles. Albeit simple, the proposed method achieves mAP 75.60% on the private testing set of CityFlow [148] without extra information. With the temporal and spatial annotation, our method further arrives the 86.07% mAP. The result surpasses the AICity Challenge champion, who also uses the temporal and spatial annotation. In a nutshell, our contributions are two-folds:

- To address the data limitation, we introduce one large-scale dataset, called VehicleNet, to borrow the strength of the public vehicle datasets, which facilitate the learning of robust vehicle features. In the experiment, we verify the feasibility and effectiveness of learning from VehicleNet.

- To leverage the multi-source vehicle images in VehicleNet, we propose a simple yet effective learning strategy, *i.e.*, the two-stage progressive learning approach. We discuss and analyze the effectiveness of the two-stage progressive learning approach. The proposed method has achieved competitive performance on the CityFlow benchmark as well as two public vehicle re-identification datasets, *i.e.*, VeRi-776 [95] and VehicleID [86].

The main content of this Chapter has been previously published in

*Zhedong Zheng, Tao Ruan, Yunchao Wei, Yi Yang, Mei Tao. "VehicleNet: Learning Robust Visual Representation for Vehicle Re-identification", IEEE Transactions on Multimedia (TMM), 2020.*

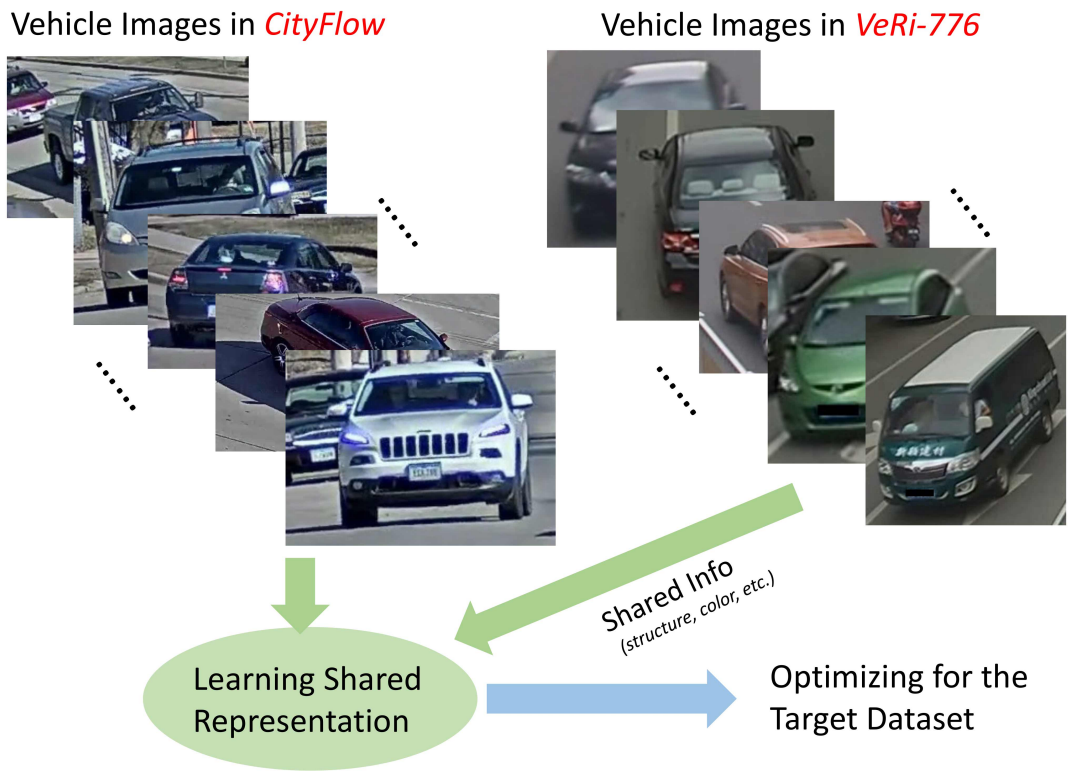


Figure 5.1 : The motivation of our vehicle re-identification method by leveraging public datasets. The common knowledge of discriminating different vehicles could be transferred to the final model.

## 5.2 Dataset Collection and Task Definition

### 5.2.1 Dataset Analysis

We involve four public datasets, *i.e.*, CityFlow [148], VeRi-776 [95], CompCar [185] and VehicleID [86] into training. It results in 434,440 training images of 31,805 classes as **VehicleNet**. Note that four public datasets are collected in different places. There are no overlapping images with the validation set or the private test set. We plot the data distribution of all four datasets in Figure 5.2. **CityFlow** [148] is one of the largest vehicle re-id datasets. There are bounding boxes of 666 vehicle identities annotated. All images are collected from 40 cameras in a realistic scenario. We follow the official training/test protocol, which results in 36,935 training images

of 333 classes and 19,342 testing images of other 333 classes. The training set is collected from 36 cameras, and test is collected from 23 cameras. There are 19 overlapping cameras. Official protocol does not provide a validation set. We therefore further split the training set into a validation set and a small training set. After the split, the training set contains 26,803 images of 255 classes, and the validation query set includes 463 images of the rest 78 classes. We deploy the original training set as the gallery of the validation set. **VeRi-776** [95] contains 49,357 images of 776 vehicles from 20 cameras. The dataset is collected in the real traffic scenario, which is close to the setting of CityFlow. The author also provides the meta data, *e.g.*, the collected time and the location. **CompCar** [185] is designed for the fine-grained car recognition. It contains 136,726 images of 1,716 car models. The author provides the vehicle bounding boxes. By cropping and ignoring the invalid bounding boxes, we finally obtain 136,713 images for training. The same car model made in different years may contain the color and shape difference. We, therefore, view the same car model produced in the different years as different classes, which results in 4,701 classes. **VehicleID** [86] consists 2211,567 images of 26,328 vehicles. The vehicle images are collected in two views, *i.e.*, frontal and rear views. Despite the limited viewpoints, the experiment shows that VehicleID also helps the viewpoint-invariant feature learning. **Other Datasets** We also review other public datasets of vehicle images in Table 5.1. Some datasets contain limited images or views, while others lack ID annotations. Therefore, we do not use these datasets, which may potentially compromise the feature learning.

### 5.2.2 Task Definition

Vehicle re-identification aims to learn a projection function $F$, which maps the input image $x$ to the discriminative representation $f_i = F(x_i)$. Usually, $F$ is decided by minimizing the following optimization function on a set of training data $X =$

Table 5.1 : Publicly available vehicle datasets. †: We view the vehicle model produced in different years as different classes, which leads to more classes. ‡: The downloaded image number is slightly different with the report number in [86].

| Datasets | # Cameras | # Images | #IDs |
|---|---|---|---|
| CityFlow [148] | 40 | 56,277 | 666 |
| VeRi-776 [95] | 20 | 49,357 | 776 |
| CompCar [185] † | n/a | 136,713 | 4,701 |
| VehicleID [86] ‡ | 2 | 221,567 | 26,328 |
| PKU-VD1 [184] | 1 | 1,097,649 | 1,232 |
| PKU-VD2 [184] | 1 | 807,260 | 1,112 |
| VehicleReID [193] | 2 | 47,123 | n/a |
| PKU-Vehicle [7] | n/a | 10,000,000 | n/a |
| StanfordCars [66] | n/a | 16,185 | 196 |
| VehicleNet | **62** | 434,440 | **31,805** |

$\{x_i\}_{i=1}^N$ with the annotated label $Y = \{y_i\}_{i=1}^N$:

$$min \sum_{i=1}^{N} loss(WF(x_i), y_i) + \alpha\Omega(F), \tag{5.1}$$

where $loss(\cdot, \cdot)$ is the loss function, $W$ is the weight of the classifier, $\Omega(F)$ is the regularization term, and $\alpha$ is the weight of the regularization.

Our goal is to leverage the augmented dataset for learning robust image representation given that the vehicle shares the common structure. The challenge is to build the vehicle representation which could fit the different data distribution among multiple datasets. Given $X^d = \{x_i^d\}_{i=1}^N$ with the annotated label $Y^d = \{y_i^d\}_{i=1,d=1}^N$, the objective could be formulated as:

$$min \sum_{d=1}^{D} \sum_{i=1}^{N} loss(WF(x_i^d), y_i^d) + \alpha\Omega(F), \tag{5.2}$$

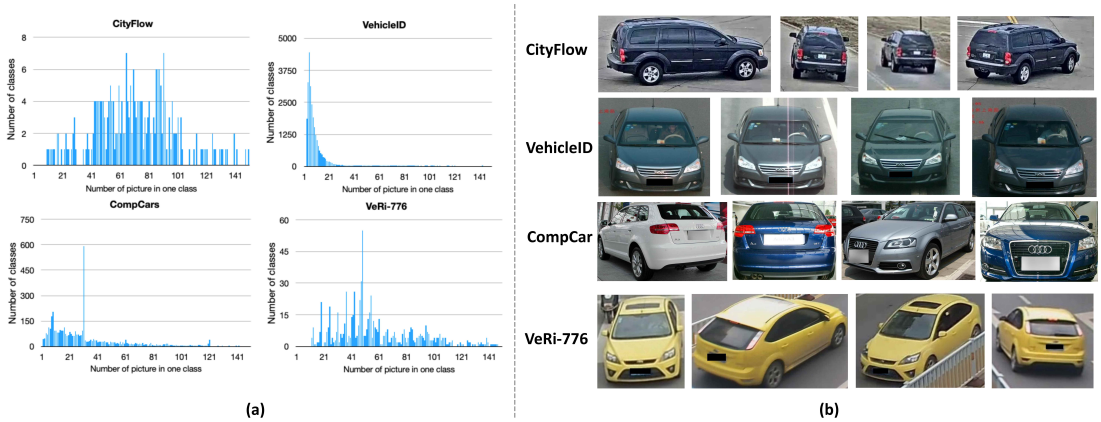


Figure 5.2 : (a) The image distribution per class in the vehicle re-id datasets, *e.g.*, CityFlow [148], VehicleID [86] , CompCar [185] and VeRi-776 [95]. We observe that the two largest datasets, *i.e.*, VehicleID and CompCars, suffer from the limited images per class. (b) Here we also provide the image samples of the four datasets. The four datasets contain different visual biases, such as illumination conditions, collection places and viewpoints.

where $D$ is the number of the augmented datasets. The loss demands $F$ could be applied to not only the target dataset but also other datasets, yielding the good scalability. In terms of the regularization term $\Omega(F)$, we adopt the common practise of weight decay as weight regularization, which prevents the weight value from growing too large and over-fits the dataset.

## 5.3 Methodology

### 5.3.1 Model Structure

**Feature Extractor.** Following the common practise in re-identification problems [95, 211], we deploy the off-the-shelf Convolutional Neural Network (CNN) model pre-trained on the ImageNet dataset [129] as the backbone. Specifically, the proposed method is scalable and could be applied to different network backbones. We have trained and evaluated the state-of-the-art structures, including ResNet-50 [42],

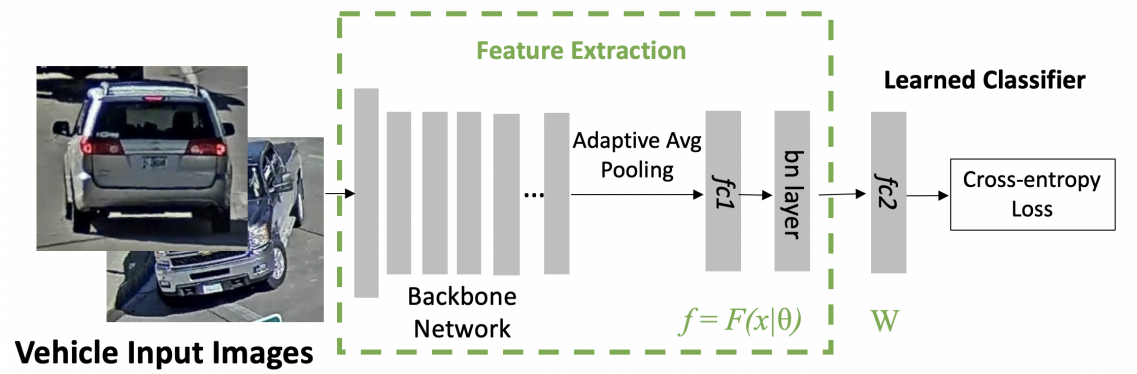


Figure 5.3 : Illustration of the model structure. We remove the original classifier of the ImageNet pre-trained model, add a new classifier and replace the average pooling with the adaptive average pooling layer. The adaptive average pooling is to squeeze the output to the pre-defined shape (*i.e.*, $1 \times 1$).

DenseNet-121 [51], SE-ResNeXt101 [48] and SENet-154 [48], in the Section 5.4. The classification layer of the pre-trained backbone model is removed, which is dedicated for image recognition on ImageNet. The original average pooling layer is replaced with the adaptive average pooling layer, and the adaptive average pooling layer outputs the mean of the input feature map in terms of the height and width channels. We add one fully-connected layer '*fc1*' of 512 dimensions and one batch normalization layer to reduce the feature dimension, followed by a fully-connected layer '*fc2*' to output the final classification prediction as shown in the Figure 5.3. The length of the classification prediction equals to the category number of the dataset. The cross-entropy loss is to penalize the wrong vehicle category prediction.

**Feature Embedding.** Vehicle re-identification is to spot the vehicle of interest from different cameras, which demands a robust representation to various visual variants, *e.g.*, viewpoints, illumination and resolution. Given the input image $x$, we intend to obtain the feature embedding $f = F(x|\theta)$. In this work, the CNN-based model contains the projection function $F$ and one linear classifier. Specifically, we regard the '*fc2*' as the conventional linear classifier with the learnable weight $W$, and

the module before the final classifier as $F$ with the learned parameter $\theta$. The output of the batch normalization layer as $f$ (see the green box in the Figure 5.3). During inference, we extract the feature embedding of query images and gallery images. The ranking list is generated according to the similarity with the query image. Given the query image, we deploy the cosine similarity, which could be formulated as $s(x_n, x_m) = \frac{f_n}{||f_n||_2} \times \frac{f_m}{||f_m||_2}$. The $||.||_2$ denotes $l^2$ norm of the corresponding feature embedding. The large similarity value indicates that the two images are highly relevant.

### 5.3.2 Two-stage Progressive Learning

The proposed training strategy contains two stages. During the first stage, we train the CNN-based model on the VehicleNet dataset and learn the general representation of the vehicle images. In particular, we deploy the widely-adopted cross-entropy loss in the recognition tasks, and the model learns to identify the input vehicle images from different classes. The loss could be formulated as:

$$L_{ce} = \sum_{i=1}^{N} -p_i \log(q_i), \tag{5.3}$$

where $p_i$ is the one-hot vector of the ground-truth label $y_i$. The one-hot vector $p_i(c) = 1$ if the index $c$ equals to $y_i$, else $p_i(c) = 0$. $q_i$ is the predicted category probability of the model, and $q_i = WF(x_i|\theta)$. Since we introduce the multi-source dataset, the cross-entropy loss could be modified to work with the multi-source data.

$$L_{ce} = \sum_{d=1}^{D} \sum_{i=1}^{N} -p_i^d \log(q_i^d), \tag{5.4}$$

where $d$ denotes the index of the public datasets in the proposed VehicleNet. Specifically, $d = 1, 2, 3, 4$ denotes the four datasets in VehicleNet, *i.e.*, CityFlow [148], VehicleID [86] , CompCar [185] and VeRi-776 [95], respectively. $p_i^d$ is the one-hot vector of $y_i^d$, and $q_i^d = WF(x_i^d|\theta)$. Note that we treat all the dataset equally, and demand the model with good scalability to data of different datasets in VehicleNet.

In the first stage, we optimize the Equation 5.4 on all the training data of VehicleNet to learn the shared representation for vehicle images. The Stage-I model is agnostic to the target environment, hence the training domain and the target domain are not fully aligned. In the second stage, we take one more step to further fine-tune the model only upon the target dataset, *e.g.*, CityFlow [148], according to the Equation 5.3. In this way, the model is further optimized for the target environment. Since only one dataset is considered in the Stage-II and the number of vehicle category is decreased, in particular, the classifier is replaced with the new $fc2$ layer with 333 classes from CityFlow. To preserve the learned knowledge, only the classification layer of the trained model is replaced. Although the new classifier is learned from scratch, attribute to the decent initial weights in the first stage, the model could converge quickly and meets the demand for quick domain adaptation. We, therefore, could stop the training at the early epoch. To summarize, we provide the training procedure of the proposed method in Algorithm 1.

**Discussion:** What are the advantages of the proposed two-stage progressive learning? First, the learned representation is more robust. In the Stage-I, we demand the model could output the discriminative representation for all of the data in the multi-source VehicleNet. The model is forced to learn the shared knowledge among the training vehicle images, which is similar to the pre-training practise in many re-id works [218, 43]. Second, the representation is also more discriminative. The first stage contains $31,805$ training classes during training. The axuiliary classes of other real vehicles could be viewed as "virtual class" as discussed in [15]. Here we provide one geometric interpretation in the Figure 5.4. After the convergence of Stage I, the cross-entropy loss pulls the data with the same label together, and pushes the data from different labels away from each other on the either side of the decision boundary. In this manner, as shown in the Figure 5.4 (right), the first stage will provide better weight initialization for the subsequent fine-tuning on the target

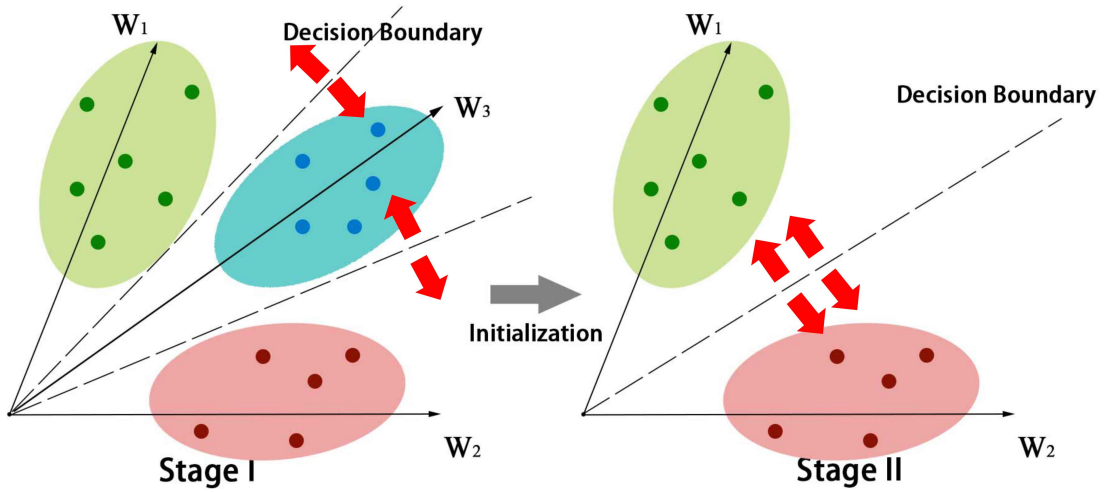


Figure 5.4 : Geometric Interpretation. Here we give a three-class sample to show our intuition. $W_i$ denotes the class weight of the final linear classifier. In this example, the third class denotes one auxiliary class, which belongs to VehicleNet but the target domain. Therefore, in the Stage-II fine-tuning, we remove the auxiliary classes, including $W_3$. The cross-entropy loss of Stage-I pulls the samples with the same label together (close to either the relative weight $W_1$, $W_2$ or $W_3$). In this way, the positive pair is closer than the negative pair, while the samples are far from the decision boundary. Stage I, therefore, leads to a decent weight initialization to be used in Stage II with a large margin from decision boundary, when we leave out the auxiliary class, *i.e.*, the third class with $W_3$, from VehicleNet.

dataset. It is because the auxiliary classes expand the decision space and the data is much far from the new decision boundary, yielding discriminative features.

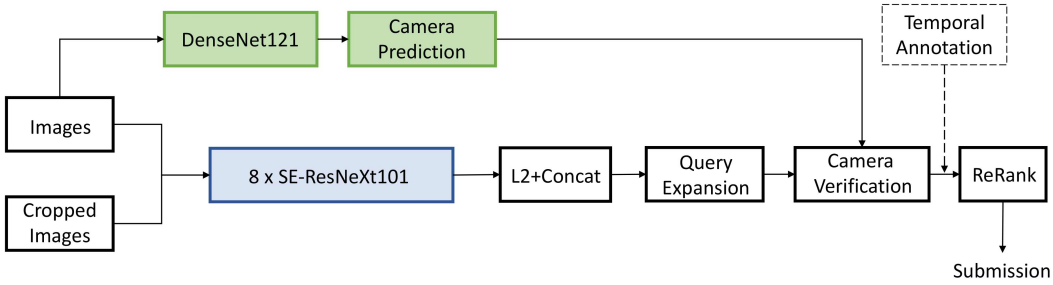


Figure 5.5 : The inference pipeline for AICity Challenge Competition. Given one input image and the corresponding cropped image via MaskRCNN [40], we extract features from the trained models, *i.e.*, 8×SE-ResNeXt101 [48]. We normalize and concatenate the features. Meanwhile, we extract the camera prediction from the camera-aware model, *i.e.*, the fine-tuned DenseNet121 [51]. Then query expansion and camera verification are applied. Finally, we utilize the re-ranking technique [220] to retrieve more positive samples.

### 5.3.3 Post-processing

Furthermore, we apply several post-processing techniques during the inference stage as shown in Figure 5.5.

**Cropped Images.** We notice that the vehicle datasets usually provide a relatively loose bounding box, which may introduce the background noise. Therefore, we re-detect vehicles with the state-of-the-art MaskRCNN [40]. For the final result, the vehicle representation is averaged between original images and cropped images, yielding more robust vehicle representations.

**Model Ensemble.** We adopt a straightforward late-fusion strategy, *i.e.*, concatenating features [218]. Given the input image $x_i$, the embedding $f_i^j$ denotes the extracted feature of $x_i$ from the $j$-th trained model. The final pedestrian descrip-

**Algorithm 1** Training Procedure of the Proposed Method

**Require:** The multi-source VehicleNet dataset $X^d = \{x_i^d\}_{i=1}^D$; The corresponding label $Y^d = \{y_i^d\}_{i=1}^D$;

**Require:** The initialized model parameter $\theta$; The first stage iteration number $T_1$ and the second stage iteration number $T_2$.

1: **for** *iteration* $= 1$ to $T_1$ **do**

2: Stage-I: Input $x_t^j$ to $F(\cdot|\theta)$, extract the prediction of the classifier, and calculate the cross-entropy loss according to Equation 5.4:

$$L_{ce} = \sum_{d=1}^{D} \sum_{i=1}^{N} -p_i^d \log(q_i^d), \tag{5.5}$$

where $p_i^d$ is the one-hot vector of $y_i^d$, and $q_i^d$ is the predict probability. $q_i^d = WF(x_i^d|\theta)$, $W$ is the final fully-connected layer, which could be viewed as a linear classifer. We update the $\theta$ and $W$ during the training.

3: **end for**

4: **for** *iteration* $= 1$ to $T_2$ **do**

5: Stage-II: We further fine-tune the trained model only on the target dataset, *e.g.*, CityFlow. The classifier is replaced with a new one, since we have less classes. We assume that CityFlow is the first dataset ($d = 1$). Thus, we could update $\theta$ upon the cross-entropy loss according to Equation 5.3:

$$L_{ce} = \sum_{i=1}^{N} -p_i^1 \log(q_i^1). \tag{5.6}$$

where $p_i^1$ is the one-hot vector of $y_i^1$ of the CityFlow dataset, and $q_i^1$ is the predict probability. $q_i^1 = W'F(x_i^1|\theta)$. We note that $W'$ is the new fully-connected layer, which is trained from scratch and different from $W$ used in the Stage-I.

6: **end for**

7: **return** $\theta$.

tor could be represented as:$f_i = [\frac{f_i^1}{||f_i^1||_2}, \frac{f_i^2}{||f_i^2||_2}, ...\frac{f_i^n}{||f_i^n||_2}]$. The $||\cdot||_2$ operator denotes $l^2$-norm, and $[\cdot]$ denotes feature concatenation.

**Query Expansion & Re-ranking.** We adopt the unsupervised clustering method, *i.e.*, DBSCAN [29] to find the most similar samples. The query feature is updated to the mean feature of the other queries in the same cluster. Furthermore, we adopt the re-ranking method [220] to refine the final result, which takes the high-confidence candidate images into consideration. In this work, our method does not modify the re-ranking procedure. Instead, the proposed method obtains discriminative vehicle features that distill the knowledge from "seeing" various cars. With better features, re-ranking is more effective.

**Camera Verification.** We utilize the camera verification to further remove some hard-negative samples. When training, we train one extra CNN model, *i.e.*, DenseNet121 [51], to recognize the camera from which the photo is taken. When testing, we extract the camera-aware features from the trained model and then cluster these features by DBSCAN [29]. In this way, we could obtain clustering centers. We applied the prior assumption that the query image and the true matches are taken in different cameras, indicating that the query images and true matches in the gallery usually belong to different camera clustering centers. Given a query image, we remove the images of the same camera cluster from candidate images.

**Temporal Annotation.** Temporal annotation can be easily obtained by recording the timestamp of which the target vehicle passes by. The prior assumption is that the vehicles usually appear once in the whole camera network, indicating that the two images with long time interval belong to two different vehicles. Given the timestamp $t$ of the query image, we filter out the image in the gallery with long interval $\tau$. As a result, we only consider the candidate images with the timestamp in $[t-\tau, t+\tau]$, which also could filter out lots of the hard-negative samples.

Table 5.2 : The Rank@1 (%) and mAP (%) accuracy with different number of training images. Here we report the results based on the validation set we splitted. † Note that we split a validation set from the training set, which leads to less training data.

| Training Datasets | # Training Images | Performance | |
|---|---|---|---|
| | | Rank@1 (%) | mAP (%) |
| CityFlow [148] † | 26,803 | 73.65 | 37.65 |
| CityFlow [148]+ VeRi-776 [95] | +49,357 | 79.48 | 43.47 |
| CityFlow [148]+ CompCar [185] | +136,713 | 83.37 | 48.71 |
| CityFlow [148]+ VehicleID [86] | +221,567 | 83.37 | 47.56 |
| VehicleNet | 434,440 | **88.77** | **57.35** |

## 5.4 Experiment

### 5.4.1 Implementation Details

For two widely-adopted public datasets, *i.e.*, VeRi-776 and VehicleID, we follow the setting in [116, 39] to conduct a fair comparison. We adopt ResNet-50 [41] as the backbone network and input images are resized to $256 \times 256$. We apply SGD optimizer with momentum of 0.9 and mini-batch size of 36. The weight decay is set to 0.0001 following the setting in [41]. The initial learning rate is set to 0.02 and is divided by a factor 10 at the 40-th epoch of the first stage and the 8-th epoch in the second stage. The total epochs of the first stage is 60 epochs, while the second-stage fine-tuning is trained with 12 epochs. **During inference, we only apply the mean feature of the image flipped horizontally, without using other post-processing approaches for two academic datasets.**

For the competition dataset, *i.e.*, CityFlow [148], we adopt one sophisticated

Table 5.3 : Comparison with the state-of-the-art methods in terms of Rank@1 (%) and mAP (%) accuracy on the VeRi-776 dataset [95] and the VehicleID dataset [86]. -: denotes the conventional hand-crafted features and *: denotes that the approach utilizes the self-designed network structure. The best results are in **bold**.

| Methods | Backbones | VeRi-776 | | VehicleID (Small) | | VehicleID (Medium) | | VehicleID (Large) | |
|---|---|---|---|---|---|---|---|---|---|
| | | mAP (%) | Rank@1 (%) | Rank@1 (%) | Rank@5 (%) | Rank@1 (%) | Rank@5 (%) | Rank@1 (%) | Rank@5 (%) |
| LOMO [80] | - | 9.78 | 23.87 | 19.74 | 32.14 | 18.95 | 29.46 | 15.26 | 25.63 |
| GoogLeNet [185] | GoogLeNet | 17.81 | 52.12 | 47.90 | 67.43 | 43.45 | 63.53 | 38.24 | 59.51 |
| FACT [95] | - | 18.73 | 51.85 | 49.53 | 67.96 | 44.63 | 64.19 | 39.91 | 60.49 |
| XVGAN [225] | * | 24.65 | 60.20 | 52.89 | 80.84 | - | - | - | - |
| SiameseVisual [131] | * | 29.48 | 41.12 | - | - | - | - | - | - |
| OIFE [169] | * | 48.00 | 65.92 | - | - | - | - | 67.0 | 82.9 |
| VAMI [226] | * | 50.13 | 77.03 | 63.12 | 83.25 | 52.87 | 75.12 | 47.34 | 70.29 |
| NuFACT [96] | * | 53.42 | 81.56 | 48.90 | 69.51 | 43.64 | 65.34 | 38.63 | 60.72 |
| FDA-Net [98] | * | 55.49 | 84.27 | - | - | 59.84 | 77.09 | 55.53 | 74.65 |
| QD-DLF [227] | * | 61.83 | 88.50 | 72.32 | 92.48 | 70.66 | 88.90 | 68.41 | 83.37 |
| GGL [93] | * | 61.7 | 89.4 | 77.1 | 92.8 | 72.7 | 89.2 | 70.0 | 87.1 |
| AAVER [61] | ResNet-50 | 58.52 | 88.68 | 72.47 | 93.22 | 66.85 | 89.39 | 60.23 | 84.85 |
| PVSS [94] | ResNet-50 | 62.62 | 90.58 | - | - | - | - | - | - |
| C2FRank [38] | GoogLeNet | - | - | 61.1 | 63.5 | 56.2 | 60.0 | 51.4 | 53.0 |
| VANet [23] | GoogLeNet | 66.34 | 89.78 | 83.26 | 95.97 | 81.11 | **94.71** | 77.21 | **92.92** |
| PAMTRI [147] | DenseNet-121 | 71.88 | 92.86 | - | - | - | - | - | - |
| SAN [116] | ResNet-50 | 72.5 | 93.3 | 79.7 | 94.3 | 78.4 | 91.3 | 75.6 | 88.3 |
| Part [39] | ResNet-50 | 74.3 | 94.3 | 78.4 | 92.3 | 75.0 | 88.3 | 74.2 | 86.4 |
| UMTS [59] | ResNet-50 | 75.9 | 95.8 | 80.9 | 87.0 | 78.8 | 84.2 | 76.1 | 82.8 |
| PVEN [105] | ResNet-50 | 79.5 | 95.6 | **84.7** | **97.0** | 80.6 | 94.5 | 77.8 | 92.0 |
| Ours (Stage-I) | ResNet-50 | 80.91 | 95.95 | 83.26 | 96.77 | 81.13 | 93.68 | 79.06 | 91.84 |
| Ours (Stage-II) | ResNet-50 | **83.41** | **96.78** | 83.64 | 96.86 | **81.35** | 93.61 | **79.46** | 92.04 |

model, *i.e.*, SE-ResNeXt101 [48] as the backbone to conduct the ablation study and report the performance. The vehicle images are resized to $384 \times 384$. Similarly, the first stage is trained with 60 epochs, and the second stage contains 12 epochs. When conducting inference on the validation set, we only apply the mean feature of the image flipped horizontally, without using other post-processing approaches. In contrast, to achieve the best results on the private test set of CityFlow, we apply all the post-processing methods mentioned in Section 5.3.3.

### 5.4.2 Quantitative Results

**Effect of VehicleNet.** To verify the effectiveness of the public vehicle data towards the model performance, we involve different vehicle datasets into training and report the results, respectively (see Table 5.2). There are two primary points as follows: First, the model performance has been improved by involving the training data of one certain datasets, either VeRi-776, CompCar or VehicleID. For instance, the model trained on CityFlow + CompCar has achieved 83.37% Rank@1 and 48.71% mAP, which surpasses the baseline of 73.65% Rank@1 and 37.65% mAP. It shows that more training data from other public datasets indeed helps the model learning the robust representation of vehicle images. Second, we utilize the proposed large-scale VehicleNet to train the model, which contains all the training data of four public datasets. We notice that there are +15.12% Rank@1 improvement from 73.65% Rank@1 to 88.77% Rank@1, and +19.70% mAP increment from 37.65% mAP to 57.35% mAP. It shows that the proposed VehicleNet has successfully "borrowed" the strength from multiple datasets and help the model learning robust and discriminative features.

**Comparison with the State-of-the-art.** We mainly compare the performance with other methods on the test sets of two public vehicle re-id datasets, *i.e.*, VeRi-776 [95] and VehicleID [86] as well as AICity Challenge [147] private test set. The

Table 5.4 : Competition results of AICity Vehicle Re-id Challenge on the private test set. Our results are in **bold**.

| Team Name | Temporal Annotation | mAP(%) |
|---|---|---|
| Baidu_ZeroOne [145] | ✓ | 85.54 |
| UWIPL [52] | ✓ | 79.17 |
| ANU [101] | ✓ | 75.89 |
| **Ours** | × | **75.60** |
| **Ours** | ✓ | **86.07** |

comparison results with other competitive methods are as follows: **VeRi-776 & VehicleID.** There are two lines of competitive methods. One line of works deploy the hand-crafted features [80, 95] or utilize the self-designed network [226, 169, 96]. In contrast, another line of works leverages the model pre-trained on ImageNet, yielding the superior performance [61, 23, 147, 39]. As shown in Table 5.3, we first evaluate the proposed approach on the VeRi-776 dataset [95]. We leave out the VeRi-776 test set from the VehicleNet to fairly compare the performance, and we deploy the ResNet-50 [41] as backbone network, which is used by most compared methods. The proposed method has achieved 83.41% mAP and 96.78% Rank@1 accuracy, which is superior to the second best method, *i.e.*, Part-based model [39] (74.3% mAP and 94.3% Rank@1) by a large margin. Meanwhile, we observe a similar result on the VehicleID dataset [86] in all three settings (Small /Medium /Large). Small, Medium and Large setting denotes different gallery sizes of 800, 1600 and 2400, respectively. The proposed method also arrives competitive results, *e.g.*, 83.64% Rank@1 of the small gallery setting, 81.35% Rank@1 of the medium gallery setting, and 79.46% Rank@1 of the large gallery setting. One competitive method, VANet [23], has achieved comparable results on VehicleID, but is inferior

to the proposed method on VeRi-776. It is because VANet introduces one extra viewpoint module, which could discriminate different viewpoints, i.e., front view and rear view. Since the VehicleID dataset only contains two views, VANet works well. In contrast, on another benchmark VeRi-776, containing 20 cameras, the proposed method is more scalable than VANet in terms of the multi-camera scenario. **AICity Challenge.** For AICity Challenge Competition (on the private test set of CityFlow [148]), we adopt a slightly different training strategy, using the large input size as well as the model ensemble. The images are resized to $384 \times 384$. We adopt the mini-batch SGD with the weight decay of 5e-4 and a momentum of 0.9. In the first stage, we decay the learning rate of 0.1 at the 40-th and 55-th epoch. We trained 32 models with different batchsizes and different learning rates. In the second stage, we fine-tune the models on the original dataset. We decay the learning rate of 0.1 at the 8-th epoch and stop training at the 12-th epoch. Finally, we select 8 best models on the validation set to extract the feature. When testing, we adopt the horizontal flipping and scale jittering, which resizes the image with the scale factors $[1, 0.9, 0.8]$ to extract features. As a result, we arrive at 75.60% mAP on the private testing set. Without extra temporal annotations, our method has already achieved competitive results (see Table 5.4). With the help of extra annotation of temporal and spatial information, we have achieved 86.07% mAP, which surpasses the champion of the AICity Vehicle Re-id Challenge 2019.

### 5.4.3 Further Evaluations and Discussion

**Effect of Two-stage Progressive Learning.** We compare the final results of the Stage I and the Stage II on the private test set of CityFlow (see Table 5.5). We do not evaluate the performance on the validation set we splitted, since we utilize all training images into fine-tuning. The model of Stage II has arrived 87.45% Rank@1 and 75.60% mAP accuracy, which has significantly surpassed the one of

Table 5.5 : The Rank@1(%) and mAP (%) accuracy with different stages on the CityFlow private test set.

| | Private Test Set | |
|---|---|---|
| | Rank@1(%) | mAP(%) |
| Stage I | 82.70 | 68.21 |
| Stage II | 87.45 | 75.60 |

Table 5.6 : Effect of different post-processing techniques on the CityFlow validation set.

| Method | Performance | | | | | |
|---|---|---|---|---|---|---|
| with Cropped Image? | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Model Ensemble? | | ✓ | ✓ | ✓ | ✓ | |
| Query Expansion? | | | ✓ | ✓ | ✓ | |
| Camera Verification? | | | | ✓ | ✓ | |
| Re-ranking? | | | | | ✓ | |
| mAP (%) | 57.35 | 57.68 | 61.29 | 63.97 | 65.97 | 74.52 |

Table 5.7 : The Rank@1 (%) and mAP (%) accuracy with different backbones on the CityFlow validation set. The best results are in **bold**.

| Backbones | ImageNet | Performance | |
|---|---|---|---|
| | Top5(%) | Rank@1 (%) | mAP (%) |
| ResNet-50 [41] | 92.98 | 77.97 | 43.65 |
| DenseNet-121 [51] | 92.14 | 83.15 | 47.17 |
| SE-ResNeXt101 [48] | 95.04 | **83.37** | **48.71** |
| SENet-154 [48] | 95.53 | 81.43 | 45.14 |

Table 5.8 : The Rank@1(%) and mAP (%) accuracy on the CityFlow validation set with two different sampling methods. Here we use the ResNet-50 backbone.

| Sampling Policy | Performance | |
|---|---|---|
| | Rank@1(%) | mAP(%) |
| Naive Sampling | 77.97 | 43.65 |
| Balanced Sampling | 76.03 | 40.09 |


Figure 5.6 : Qualitative image search results using the vehicle query images from the CityFlow dataset. We select the four query images from different viewpoints. The results are sorted from left to right according to the similarity score. The true-matches are in green, when the false-matches are in red.

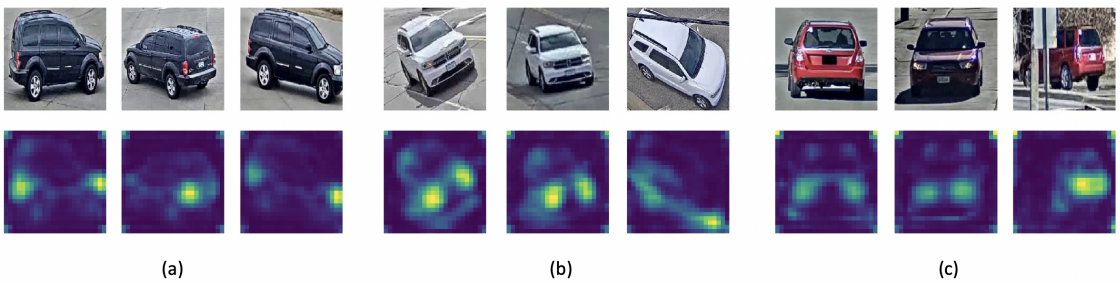


Figure 5.7 : Visualization of the activation heatmap in the learned model on VehicleNet. The vehicle images in every subfigure (a)-(c) are from the same vehicle ID. Noted that there do exist strong response values at the regions containing discriminative details, such as headlights and tire types.

Stage I +7.39% mAP and +4.75% Rank@1. It verifies the effectiveness of the two-stage learning. In the Stage I, the target training set, *i.e.*, CityFlow, only occupy 6% of VehicleNet. The learned model, therefore, is sub-optimal for the target environment. To further optimize the model for CityFlow, the second stage fine-tuning helps to minor the gap between VehicleNet and the target set, yielding better performance. We also observe similar results on the other two datasets, *i.e.*, VeRi-776 and VehicleID. As shown in the last two row of Table 5.3, the Stage-II fine-tuning could further boost the performance. For instance, the proposed method has achieved +2.50% mAP and +0.83% Rank@1 improvement on VeRi-776. We compare the two-stage learning strategy with the domain adaption policy, which is usually based on style transferring. Specifically, we apply the prevailing CycleGAN [229] to change the style of data in VehicleNet to VeRi-776. We observe that CycleGAN could successfully change the vehicle style. However, CycleGAN introduces some unrealistic noise. As shown in Table 5.9, the style transferring method is inferior to the proposed two-stage learning strategy. We speculate that it is due to the generation noise by CycleGAN. Besides, training CycleGAN costs extra time, which may be not ideal for the fast domain adaptation.

Table 5.9 : Comparison with other complementary methods on VeRi-776.

| Method | Rank@1(%) | mAP(%) |
|---|---|---|
| *w* CycleGAN data | 92.91 | 75.23 |
| Stage I | 95.95 | 80.91 |
| Stage II | 96.78 | 83.41 |
| Stage II + PCB [142] | 97.26 | 83.54 |

**Effect of Part-based Method Fusion.** The proposed method has the potential to fuse with other competitive methods. We select the second best method [116] on

VeRi-776 to verify the potential of the proposed method. [116] utilizes one similar policy as PCB [142] to split the feature map horizontally into 4 parts. As shown in Table 5.9, ours + PCB can take one step further, yielding 97.26% Rank@1 and 83.54% mAP.

**Effect of Post-processing.** Here we provide the ablation study of post-processing techniques on the validation set of CityFlow (see Table 5.6). When applying the augmentation with cropped images, model ensemble, query expansion, camera verification and re-ranking, the performance gradually increases, which verifies the effectiveness of post-processing methods.

**Effect of Different Backbones.** We observe that different backbones may lead to different results. As shown in Table 5.7, SE-ResNeXt101 [48] arrives the best performance with 83.37 Rank@1 and 48.71% mAP on the validation set of the CityFlow dataset. We speculate that it is tricky to optimize some large-scale neural networks due to the problem of gradient vanishing. For instance, we do not achieve a better result (45.14% mAP) with SENet-154 [48], which preforms better than SE-ResNeXt101 [48] on ImageNet [25]. We hope this observation could help the further study of the model backbone selection in terms of the re-identification task.

**Effect of Sampling Policy.** Since we introduce more training data in the first stage, the data sampling policy has a large impact on the final result. We compare two sampling policies. The naive method is to sample every image once in every epoch. Another method is called balanced sampling policy. The balanced sampling is to sample the images of different class with equal possibility. As shown in Table 5.8, the balanced sampling harms the result. We speculate that the long-tailed data distribution (see Figure 5.2) makes the balanced sampling have more chance to select the same image in the classes with fewer images. Thus the model is prone to over-fit the class with limited samples, which compromise the final performance.

Therefore, we adopt the naive sampling policy.

**Visualization of Vehicle Re-id Results.** As shown in Figure 5.6, we provide the qualitative image search results on CityFlow. We select the four query images from different viewpoints, *i.e.*, the front view, the overhead view, the rear view and the side view. The proposed method has successfully retrieved the relevant results in the top-5 of the ranking list.

**Visualization of Learned Heatmap.** Following [216, 7], we utilize the network activation before the pooling layer to visualize the attention of the learned model. For instance, given one middle-level feature of $14 \times 14 \times 2048$, we aggregate the activation of all channels via summation, resulting one feature of $14 \times 14$. Then we normalize the feature to [0,1], and map the value to the corresponding heatmap color. The generation code is avaiable at *. As shown in Figure 5.7, the trained model has strong response values at the regions containing discriminative details, such as headlights and tire types. In particular, despite different viewpoints, the model could focus on the salient areas, yielding the viewpoint-invariant feature.

**Model Convergence.** As shown in Figure 5.8 (left), despite a large number of training classes, *i.e.*, $31,805$ categories in VehicleNet, the model could converge within 60 epochs. As discussed, the first stage provides a decent weight initialization for fine-tuning in the second stage. Therefore, Stage-II training converges quickly within 12 epochs (see Figure 5.8 (right)).

**Time Cost.** The Stage-I training costs about 30 hours on the whole VehicleNet with $3 \times$ Nvidia 2080TI. The Stage-II training costs about 1.5 hours for fine-tuning.

*https://github.com/layumi/Person_reID_baseline_pytorch/blob/dev/visual_heatmap.py

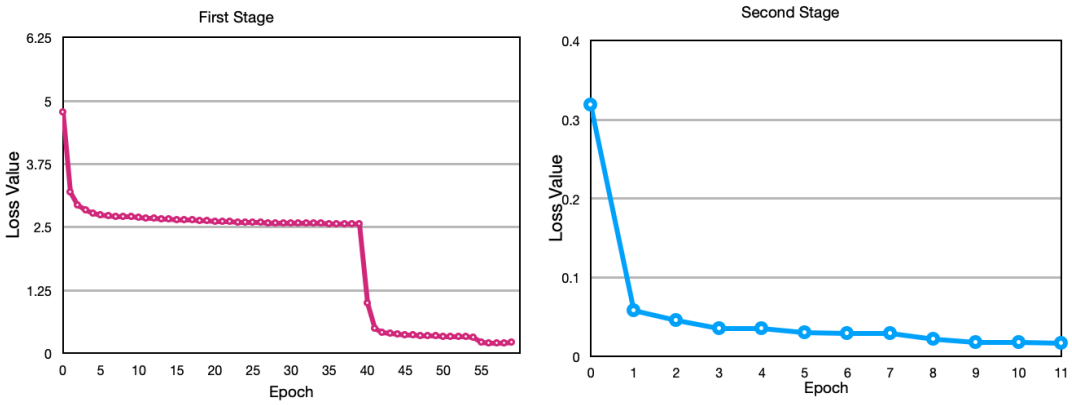


Figure 5.8 : The training losses of the two stages. Due to the large-scale data and classes, the first stage (left) takes more epochs to converge. Attribute to the trained weight of the first stage, the second stage (right) converge early.

## 5.5 Summary

In this chapter, we intend to address two challenges in the context of vehicle re-identification, *i.e.*, the lack of training data, and how to harness multiple public datasets. To address the data limitation, we build a large-scale dataset called VehicleNet. To learn the robust feature, we propose a simple yet effective approach, called two-stage progressive learning. We achieve 86.07% mAP accuracy in AICity19 Challenge and competitive performance on two other public datasets, *i.e.*, VeRi-776 and VehicleID. In the future, we will try two data collection methods to further improve the work. 1). One method is to collect data from the search engine, *i.e.*, Google, to enlarge the dataset. The existing works [65, 210] show that a few noise annotations usually do not compromise the model training. 2). The other way is to generate the synthetic data by either GAN [36] or 3D-models [187], to further explore the robust representation learning. Besides, we will explore weakly supervised learning approaches [196, 195, 106] to fully take advantage of unlabeled data.

# Chapter 6

# Joint Discriminative and Generative Learning

## 6.1 Introduction

Person re-identification (re-id) aims to establish identity correspondences across different cameras. It is often approached as a metric learning problem [207], where one seeks to retrieve images containing the person of interest from non-overlapping cameras given a query image. This is challenging in the sense that images captured by different cameras often contain significant intra-class variations caused by the changes in background, viewpoint, human pose, etc. As a result, designing or learning representations that are robust against intra-class variations as much as possible has been one of the major targets in person re-id.

Convolutional neural networks (CNNs) have recently become increasingly predominant choices in person re-id thanks to their strong representation power and the ability to learn invariant deep embeddings. Current state-of-the-art re-id methods widely formulate the tasks as deep metric learning problems [216, 43], or use classification losses as the proxy targets to learn deep embeddings [207, 77, 142, 180, 218, 148]. To further reduce the influence from intra-class variations, a number of existing methods adopt part-based matching or ensemble to explicitly align and compensate the variations [136, 204, 172, 138, 218].

Another possibility to enhance robustness against input variations is to let the re-id model potentially "see" these variations (particularly intra-class variations) during training. With recent progress in the generative adversarial networks (GANs) [36], generative models have become appealing choices to introduce additional augmented


Figure 6.1 : Examples of generated images on Market-1501 by switching appearance or structure codes. Each row and column corresponds to different appearance and structure.

Table 6.1 : Description of the information encoded in the latent appearance and structure spaces.

| Appearance Space | Structure Space |
|---|---|
| clothing/shoes color, texture and style, other id-related cues, etc. | body size, hair, carrying, pose, background, position, viewpoint, etc. |

data for free [217]. Despite the different forms, the general considerations behind these methods are "realism": generated images should possess good qualities to close the domain gap between synthesized scenarios and real ones; and "diversity": generated images should contain sufficient diversity to adequately cover unseen variations. Within this context, some prior works have explored unconditional GANs and human pose conditioned GANs [217, 55, 118, 33, 87] to generate pedestrian images to improve re-id learning. However, a common issue behind these methods is that their generative pipelines are typically presented as standalone models, which are relatively separate from the discriminative re-id models. Therefore, the optimization target of a generative module may not be well aligned with the re-id task, limiting the gain from generated data.

In light of the above observation, we propose a learning framework that jointly couples discriminative and generative learning in a unified network called **DG-Net**. Our strategy towards achieving this goal is to introduce a generative module, of which encoders decompose each pedestrian image into two latent spaces: an **appearance** space that mostly encodes appearance and other identity related semantics; and a **structure** space that encloses geometry and position related structural information as well as other additional variations. We refer to the encoded fea-

tures in the space as "codes". The properties captured by the two latent spaces are summarized in Table 6.1. The appearance space encoder is also shared with the discriminative module, serving as a re-id learning backbone. This design leads to a single unified framework that subsumes these interactions between generative and discriminative modules: (1) the generative module produces synthesized images that are taken to refine the appearance encoder online; (2) the encoder, in turn, influences the generative module with improved appearance encoding; and (3) both modules are jointly optimized, given the shared appearance encoder.

We formulate the image generation as switching the appearance or structure codes between two images. Given any pairwise images with the same/different identities, one is able to generate realistic and diverse intra/cross-id composed images by manipulating the codes. An example of such composed image generation on Market-1501 [206] is shown in Figure 6.1. Our design of the generative pipeline not only leads to high-fidelity generation, but also yields substantial diversity given the combinatorial compositions of existing identities. Unlike the unconditional GANs [217, 55], our method allows more controllable generation with better quality. Unlike the pose-guided generations [118, 33, 87], our method does not require any additional auxiliary data, but takes the advantage of existing intra-dataset pose variations as well as other diversities beyond pose.

This generative module design specifically serves for our discriminative module to better make use of the generated data. For one pedestrian image, by keeping its appearance code and combining with different structure codes, we can generate multiple images that remain clothing and shoes but change pose, viewpoint, background, etc. As demonstrated in each row of Figure 6.1, these images correspond to the same clothing dressed on different people. To better capture such composed cross-id information, we introduce the "primary feature learning" via a dynamic soft labeling strategy. Alternatively, we can keep one structure code and combine

with different appearance codes to produce various images, which maintain the pose, background and some identity related fine details but alter clothes and shoes. As shown in each column of Figure 6.1, these images form an interesting simulation of the same person wearing different clothes and shoes. This creates an opportunity for further mining the subtle identity attributes that are independent of clothing, such as carrying, hair, body size, etc. Thus, we propose the complementary "fine-grained feature mining" to learn additional subtle identity properties.

To our knowledge, this work provides the first framework that is able to end-to-end integrate discriminative and generative learning in a single unified network for person re-id. Extensive qualitative and quantitative experiments show that our image generation compares favorably against the existing ones, and more importantly, our re-id accuracy consistently outperforms the competing algorithms by large margins on several benchmarks.

The main content of this Chapter has been previously published in

*Zhedong Zheng, Xiaodong Yang, Zhiding Yu, Liang Zheng, Yi Yang, Jan Kautz. "Joint Discriminative and Generative Learning for Person Re-identification", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. (Oral)*

## 6.2 Methodology

As illustrated in Figure 6.2, DG-Net tightly couples the generative module for image generation and the discriminative module for re-id learning. We introduce two image mappings: self-identity generation and cross-identity generation to synthesize high-quality images that are online fed into re-id learning. Our discriminative module involves primary feature learning and fine-grained feature mining, which are co-designed with the generative module to better leverage the generated data.

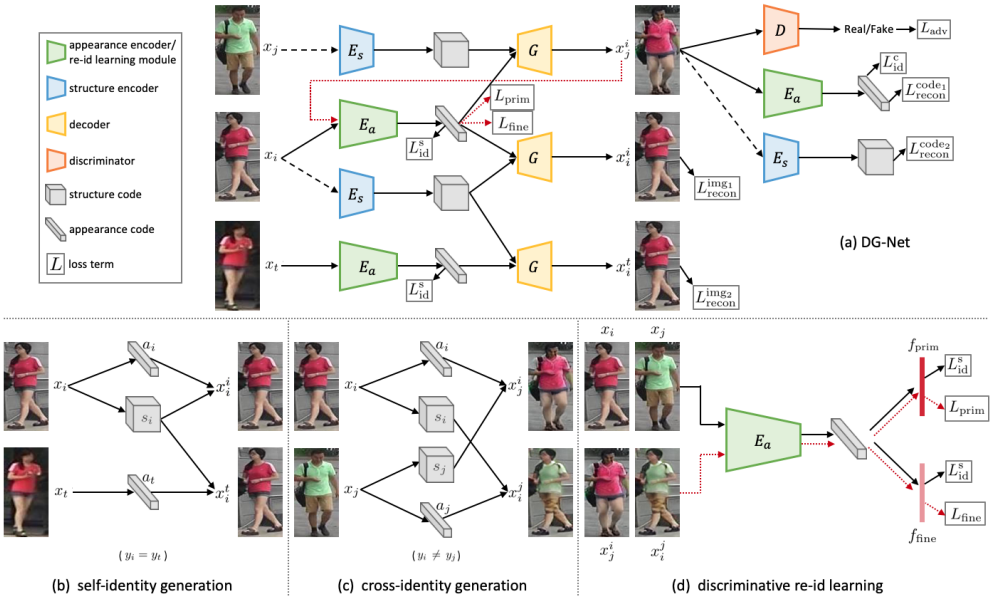


Figure 6.2 : A schematic overview of DG-Net. (a) Our discriminative re-id learning module is embedded in the generative module by sharing appearance encoder $E_a$. A dash black line denotes the input image to structure encoder $E_s$ is converted to gray. The red line indicates the generated images are online fed back to $E_a$. Two objectives are enforced in the generative module: (b) self-identity generation by the same input identity and (c) cross-identity generation by different input identities. (d) To better leverage generated data, the re-id learning involves primary feature learning and fine-grained feature mining.

### 6.2.1 Generative Module

**Formulation.** We denote the real images and identity labels as $X = \{x_i\}_{i=1}^N$ and $Y = \{y_i\}_{i=1}^N$, where $N$ is the number of images, $y_i \in [1, K]$ and $K$ indicates the number of classes or identities in the dataset. Given two real images $x_i$ and $x_j$ in the training set, our generative module generates a new pedestrian image by swapping the appearance or structure codes of the two images. As shown in Figure 6.2, the generative module consists of an appearance encoder $E_a : x_i \rightarrow a_i$, a structure encoder $E_s : x_j \rightarrow s_j$, a decoder $G : (a_i, s_j) \rightarrow x_j^i$, and a discriminator

$D$ to distinguish between generated images and real ones. In the case $i = j$, the generator can be viewed as an auto-encoder, so $x_i^i \approx x_i$. Note: for generated images, we use superscript to denote the real image providing appearance code and subscript to indicate the one offering structure code, while real images only have subscript as image index. Compared to the appearance code $a_i$, the structure code $s_j$ maintains more spatial resolution to preserve geometric and positional properties. However, this may result in a trivial solution for $G$ to only use $s_j$ but ignore $a_i$ in image generation since decoders tend to rely on the feature with more spatial information. In practice, we convert input images of $E_s$ into gray-scale to drive $G$ to leverage both $a_i$ and $s_j$. We enforce the two objectives for the generative module: (1) self-identity generation to regularize the generator and (2) cross-identity generation to make generated images controllable and match real data distribution.

**Self-identity generation.** As illustrated in Figure 6.2(b), given an image $x_i$, the generative module first learns how to reconstruct $x_i$ from itself. This simple self-reconstruction task serves as an important regularization role to the whole generation. We reconstruct the image using the pixel-wise $\ell_1$ loss:

$$L_{\text{recon}}^{\text{img}_1} = \mathbb{E}[\|x_i - G(a_i, s_i)\|_1]. \tag{6.1}$$

Based on the assumption that the appearance codes of the same person in different images are close, we further propose another reconstruction task between any two images of the same identity. In other words, the generator should be able to reconstruct $x_i$ through an image $x_t$ with the same identity $y_i = y_t$:

$$L_{\text{recon}}^{\text{img}_2} = \mathbb{E}[\|x_i - G(a_t, s_i)\|_1]. \tag{6.2}$$

This same-identity but cross-image reconstruction loss encourages the appearance encoder to pull appearance codes of the same identity together so that intra-class feature variations are reduced. In the meantime, to force the appearance codes of different images to stay apart, we use identification loss to distinguish different

identities:

$$L_{\text{id}}^{\text{s}} = \mathbb{E}[-\log(p(y_i|x_i))], \tag{6.3}$$

where $p(y_i|x_i)$ is the predicted probability that $x_i$ belongs to the ground-truth class $y_i$ based on its appearance code.

**Cross-identity generation.** Different from self-identity generation that works with image reconstruction using the same identity, cross-identity generation focuses on image generation with different identities. In this case, there is no pixel-level ground-truth supervision. Instead, we introduce the latent code reconstruction based on appearance and structure codes to control such image generation. As shown in Figure 6.2(c), given two images $x_i$ and $x_j$ of different identities $y_i \neq y_j$, the generated image $x_j^i = G(a_i, s_j)$ is required to retain the information of appearance code $a_i$ from $x_i$ and structure code $s_j$ from $x_j$, respectively. We should then be able to reconstruct the two latent codes after encoding the generated image:

$$L_{\text{recon}}^{\text{code}_1} = \mathbb{E}[\|a_i - E_a(G(a_i, s_j))\|_1], \tag{6.4}$$

$$L_{\text{recon}}^{\text{code}_2} = \mathbb{E}[\|s_j - E_s(G(a_i, s_j))\|_1]. \tag{6.5}$$

Similar for self-identity generation, we also enforce identification loss on the generated image based on its appearance code to keep the identity consistency:

$$L_{\text{id}}^{\text{c}} = \mathbb{E}[-\log(p(y_i|x_j^i))], \tag{6.6}$$

where $p(y_i|x_j^i)$ is the predicted probability of $x_j^i$ belonging to the ground-truth class $y_i$ of $x_i$, the image that provides appearance code in generating $x_j^i$. Additionally, we employ adversarial loss to match the distribution of generated images to the real data distribution:

$$L_{\text{adv}} = \mathbb{E}[\log D(x_i) + \log(1 - D(G(a_i, s_j))]. \tag{6.7}$$

**Discussion.** By using the proposed generation mechanism, we enable the generative module to learn appearance and structure codes with explicit and comple-

mentary meanings and generate high-quality pedestrian images based on the latent codes. This largely eases the generation complexity. In contrast, the previous methods [217, 55, 118, 33, 87] have to learn image generation either from random noise or managing the pose factor only, which is hard to manipulate the outputs and inevitably introduces artifacts. Moreover, due to using the latent codes, the variants in our generated images are explainable and constrained in the existing contents of real images, which also ensures the generation realism. In theory, we can generate $O(N \times N)$ different images by sampling various image pairs, resulting in a much larger online generated training sample pool than the ones with $O(2 \times N)$ images offline generated in [217, 55, 118].

### 6.2.2 Discriminative Module

Our discriminative module is embedded in the generative module by sharing the appearance encoder as the backbone for re-id learning. In accordance with the images generated by switching either appearance or structure codes, we propose the primary feature learning and fine-grained feature mining to better take advantage of the online generated images. Since the two tasks focus on different aspects of generated images, we branch out two lightweight headers on top of the appearance encoder for the two types of feature learning, as illustrated in Figure 6.2(d).

**Primary feature learning.** It is possible to treat the generated images as training samples similar to the existing work [217, 55, 118]. But the inter-class variations in the cross-id composed images motivate us to adopt a teacher-student type supervision with dynamic soft labeling. We use a teacher model to dynamically assign a soft label to $x_j^i$, depending on its compound appearance and structure from $x_i$ and $x_j$. The teacher model is simply a baseline CNN trained with identification loss on the original training set. To train the discriminative module for primary feature learning, we minimize the KL divergence between the probability distribution

$p(x_j^i)$ predicted by the discriminative module and the probability distribution $q(x_j^i)$ predicted by the teacher:

$$L_{\text{prim}} = \mathbb{E}[-\sum_{k=1}^{K} q(k|x_j^i) \log(\frac{p(k|x_j^i)}{q(k|x_j^i)})], \tag{6.8}$$

where $K$ is the number of identities. In comparison with the fixed one-hot label [118, 232] or static smoothing label [217], this dynamic soft labeling fits better in our case, as each synthetic image is formed by the visual contents from two real images. In the experiments, we show that a simple baseline CNN serving as the teacher model is reliable to provide the dynamic labels and improve the performance.

**Fine-grained feature mining.** Beyond the direct usage of generated data for learning primary features, an interesting alternative, made possible by our specific generation pipeline, is to simulate the change of clothing for the same person, as shown in each column of Figure 6.1. When training on images organized in this manner, the discriminative module is forced to learn the fine-grained id-related attributes (such as hair, hat, bag, body size, and so on) that are independent to clothing. We view the images generated by one structure code combining with different appearance codes as the same class as the real image providing the structure code. To train the discriminative module for fine-grained feature mining, we enforce identification loss on this particular categorizing:

$$L_{\text{fine}} = \mathbb{E}[-\log(p(y_j|x_j^i))]. \tag{6.9}$$

This loss imposes additional identity supervision to the discriminative module in a multi-tasking way. Moreover, unlike the previous works using manually labeled pedestrian attributes [84, 137, 164], our approach performs automatic fine-grained attribute mining by leveraging on the synthetic images. Furthermore, compared to the hard sampling policy applied in [43, 127], there is no need to explicitly search for the hard training samples that usually possess fine-grained details, since our

discriminative module learns to attention on the subtle identity properties through this fine-grained feature mining.

**Discussion.** We argue that our high-quality synthetic images, in nature, can be viewed as "inliers" (contrary to "outliers"), as our generated images maintain and recompose the visual contents from real data. Via the above two feature learning tasks, our discriminative module makes specific use of the generated data in line with the way how we manipulate the appearance and structure codes. Instead of using a single supervision as in almost all previous methods [217, 55, 118], we treat the generated images in two different perspectives through the primary feature learning and fine-grained feature mining, where the former focuses on the structure-invariant clothing information and the latter attentions to the appearance-invariant structural cues.

### 6.2.3 Optimization

We jointly train the appearance and structure encoders, decoder, and discriminator to optimize the total objective, which is a weighted sum of the following losses:

$$\begin{aligned} L_{\text{total}}(E_a, E_s, G, D) = \lambda_{\text{img}} L_{\text{recon}}^{\text{img}} + L_{\text{recon}}^{\text{code}} + \\ L_{\text{id}}^{\text{s}} + \lambda_{\text{id}} L_{\text{id}}^{\text{c}} + L_{\text{adv}} + \lambda_{\text{prim}} L_{\text{prim}} + \lambda_{\text{fine}} L_{\text{fine}}, \end{aligned} \tag{6.10}$$

where $L_{\text{recon}}^{\text{img}} = L_{\text{recon}}^{\text{img}_1} + L_{\text{recon}}^{\text{img}_2}$ is the image reconstruction loss in self-identity generation, $L_{\text{recon}}^{\text{code}} = L_{\text{recon}}^{\text{code}_1} + L_{\text{recon}}^{\text{code}_2}$ is the latent code reconstruction loss in cross-identity generation, $\lambda_{\text{img}}$, $\lambda_{\text{id}}$, $\lambda_{\text{prim}}$, and $\lambda_{\text{fine}}$ are weights to control the importance of related loss terms. Following the common practice in image-to-image translations [228, 71, 54], we use a large weight $\lambda_{\text{img}} = 5$ for the image reconstruction loss. Since the quality of cross-id generated images is not great at the beginning, the identification loss $L_{\text{id}}^{\text{c}}$ may make the training unstable, so we set a small weight $\lambda_{\text{id}} = 0.5$.

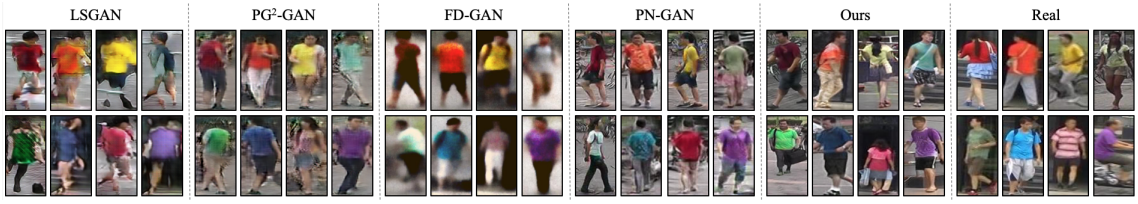


Figure 6.3 : Comparison of the generated and real images on Market-1501 across the different methods including LSGAN [104], PG$^2$-GAN [102], FD-GAN [33], PN-GAN [118], and our approach. This figure is best viewed when zoom in. Please attention to both foreground and background of the images.

We fix the two weights during the whole training process in all experiments. We do not involve the discriminative feature learning losses $L_{\text{prim}}$ and $L_{\text{fine}}$ until the generation quality is stable. As an example, we add in the two losses after 30K iterations on Market-1501, then linearly increase $\lambda_{\text{prim}}$ from 0 to 2 in 4K iterations and set $\lambda_{\text{fine}} = 0.2\lambda_{\text{prim}}$. See more details on how to determine the weights in Section 6.3.3. Similar to the alternative updating policy for GANs, in the cross-identity generation as shown in Figure 6.2(a), we alternatively train $E_a$, $E_s$ and $G$ before the generated image and $E_a$, $E_s$ and $D$ after the generated image.

## 6.3 Experiment

We evaluate the proposed approach following standard protocols on three benchmark datasets: Market-1501 [206], DukeMTMC-reID [126, 217], and MSMT17 [171]. We qualitatively and quantitatively compare DG-Net with state-of-the-art methods on both generative and discriminative results. Extensive experiments demonstrate that DG-Net produces more realistic and diverse images, and meanwhile, consistently outperforms the most recent competing algorithms by large margins on re-id accuracy across all benchmarks.

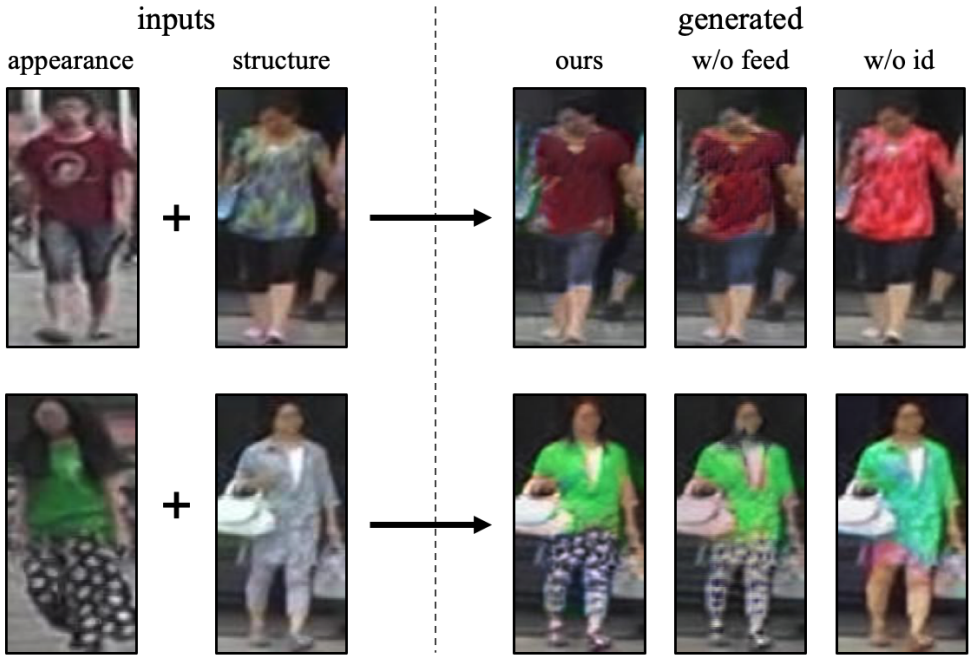


Figure 6.4 : Comparison of the generated images by our full model, removing online feeding (w/o feed), and further removing identity supervision (w/o id).

### 6.3.1 Implementation Details

Our network is implemented in PyTorch. In the following, we use channel×height×width to indicate the size of feature maps. (**i**) $E_a$ is based on ResNet50 [41] pre-trained on ImageNet [25], and we remove its global average pooling layer and fully-connected layer then append an adaptive max pooling layer to output the appearance code $a$ in $2048 \times 4 \times 1$. It is mapped to primary feature $f_{\text{prim}}$ and fine-grained feature $f_{\text{fine}}$, both are 512-dim vectors, through two fully-connected layers. (**ii**) $E_s$ is a shallow network that outputs the structure code $s$ in $128 \times 64 \times 32$. It consists of four convolutional layers followed by four residual blocks [41]. (**iii**) $G$ processes $s$ by four residual blocks and four convolutional layers. As in [54] every residual block contains two adaptive instance normalization layers [53], which integrate in $a$ as scale and bias parameters. (**iv**) $D$ follows the popular multi-scale PatchGAN [58]. We employ discriminators on the three different input image scales: $64 \times 32$, $128 \times 64$,

Figure 6.5 : Example of image generation by linear interpolation between two appearance codes.

and $256 \times 128$. We also apply the gradient punishment [107] when updating $D$ to stabilize training. (**v**) For training, all input images are resized to $256 \times 128$. Similar to the previous deep re-id models [207], SGD is used to train $E_a$ with learning rate 0.002 and momentum 0.9. We apply Adam [63] to optimize $E_s$, $G$ and $D$, and set learning rate to 0.0001, and $(\beta_1, \beta_2) = (0, 0.999)$. (**vi**) At test time, our re-id model only involves $E_a$ (along with two lightweight headers), which is of a comparable network size to most methods using ResNet50 as the backbone. We concatenate $f_{\text{prim}}$ and $f_{\text{fine}}$ into a 1024-dim vector as the final pedestrian representation. More architecture details can be found in the appendix.

### 6.3.2 Generative Evaluations

**Qualitative evaluations.** We first qualitatively compare DG-Net with its two variants that ablate online feeding and identity supervision. As shown in Figure 6.4, without online feeding generated images to appearance encoder, the model suffers from blurry edges and undesired textures. If further removing identity supervision, the image quality is unsatisfying as the model fails to produce the accurate clothing color or style. This clearly shows that our joint discriminative learning is beneficial to the image generation.


Figure 6.6 : Examples of our generated images by swapping appearance or structure codes on the three datasets. All images are sampled from the test sets.

Next we compare our full model with other generative approaches, including one unconditional GAN (LSGAN [104]) and three open-source conditional GANs (PG$^2$-GAN [102], PN-GAN [118] and FD-GAN [33]). As compared in Figure 6.3, the images generated by LSGAN have severe artifacts and duplicated patterns. FD-GAN are prone to generate very blurry images, which largely deteriorate the realism. PG$^2$-GAN and PN-GAN, both conditioned on pose, generate relatively good visual results, but still contain visible blurs and artifacts especially in background. In comparison, our generated images are more realistic and close to the real in both foreground and background.

To better understand the learned appearance space, which is the foundation for our pedestrian representations, we perform a linear interpolation between two appearance codes and generate the corresponding images as shown in Figure 6.5. These interpolation results verify the continuity in the appearance space, and show that our model is able to generalize in the space instead of simply memorizing trivial visual information. As a complementary study, we also generate images by linearly interpolating between two structure codes while keeping the appearance code intact. See more discussions regarding this study in the appendix. We then demonstrate our generation results on the three benchmarks in Figure 6.6, where DG-Net is found

Table 6.2 : Comparison of FID (lower is better) and SSIM (higher is better) to evaluate realism and diversity of the real and generated images on Market-1501.

| Methods | Realism (FID) | Diversity (SSIM) |
|---|---|---|
| Real | 7.22 | 0.350 |
| LSGAN [104] | 136.26 | - |
| $PG^2$-GAN [102] | 151.16 | - |
| PN-GAN [118] | 54.23 | 0.335 |
| FD-GAN [33] | 257.00 | 0.247 |
| Ours | **18.24** | **0.360** |

to be able to consistently generate realistic and diverse images across the different datasets.

**Quantitative evaluations.** Our qualitative observations above are confirmed by the quantitative evaluations. We use two metrics: Fréchet Inception Distance (FID)[44] and Structural SIMilarity (SSIM) [170] to measure realism and diversity of generated images, respectively. FID measures how close the distribution of generated images is to the real. It is sensitive to visual artifacts and thus indicates the realism of generated images. For the identity conditioned generation, we apply SSIM to compute intra-class similarity, which can be used to reflect the generation diversity. As shown in Table 6.2, our approach significantly outperforms other methods on both realism and diversity, suggesting the high quality of our generated images. Remarkably, we obtain a higher SSIM than the original training set thanks to the various poses, carryings, backgrounds, etc. introduced by switching structure codes.

**Limitation.** We notice that due to data bias in the original training set, our

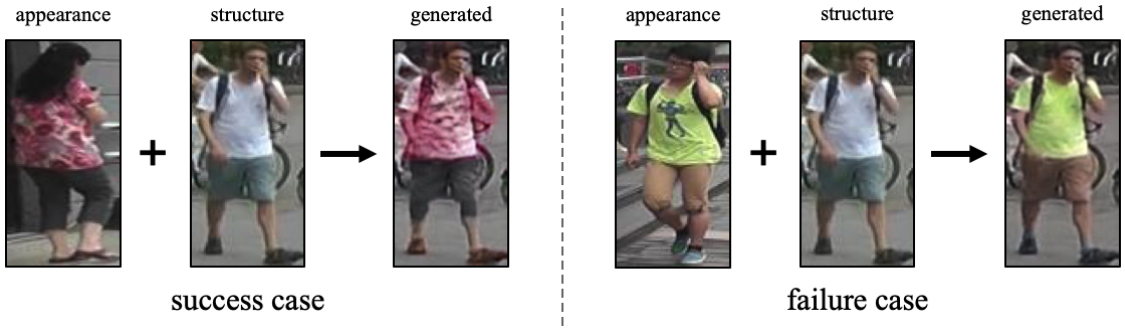


Figure 6.7 : Comparison of success and failure cases in our image generation. In the failure case, the logo on t-shirt of the original image is missed in the synthetic image.

generative module tends to learn the regular textures (e.g., stripes and dots) but ignores some rare patterns (e.g., logos on shirts), as shown in Figure 6.7.

### 6.3.3 Discriminative Evaluations

**Ablation studies.** We first study the contributions of primary feature and fine-grained feature in Table 6.3. We train ResNet50 with identification loss on each original training set as the baseline. It also serves as the teacher model in primary feature learning to perform dynamic soft labeling on the generated images. Our primary feature is found to largely improve over the baseline. Notably, the fine-grained feature without using important appearance information but only considering subtle id-related cues already achieves impressive accuracy. By combining the two features, we can further improve the performance, which substantially outperforms the baseline by 6.1% for Rank@1 and 12.4% for mAP on average of the three datasets. We then evaluate the two features independently learned after our synthetic images are offline generated. This results in an 84.4% mAP on Market-1501, inferior to the 86.0% mAP of the end-to-end training, suggesting that our joint generative training is beneficial to the re-id learning.

**Influence of hyper-parameters.** Here we show how to set the re-id learning

Table 6.3 : Comparison of baseline, primary feature, fine-grained feature, and their combination on the three datasets.

| Methods | Market-1501 | | DukeMTMC-reID | | MSMT17 | |
|---|---|---|---|---|---|---|
| | Rank@1 | mAP | Rank@1 | mAP | Rank@1 | mAP |
| Baseline | 89.6 | 74.5 | 82.0 | 65.3 | 68.8 | 36.2 |
| $f_{\text{prim}}$ | 94.0 | 84.4 | 85.6 | 72.7 | 76.0 | 49.7 |
| $f_{\text{fine}}$ | 91.6 | 75.3 | 78.7 | 61.2 | 71.5 | 43.5 |
| $f_{\text{prim}}, f_{\text{fine}}$ | **94.8** | **86.0** | **86.6** | **74.8** | **77.2** | **52.3** |

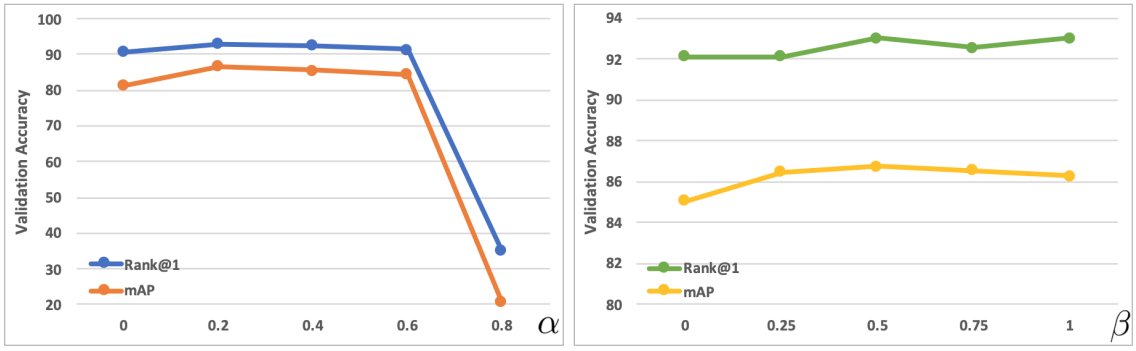


Figure 6.8 : Analysis of the re-id learning related hyper-parameters $\alpha$ and $\beta$ to balance primary and fine-grained features in training (left) and testing (right).

related weights: one is $\alpha$, the ratio between $\lambda_{\text{fine}}$ and $\lambda_{\text{prim}}$ to control the importance of $L_{\text{fine}}$ and $L_{\text{prim}}$ in training; the other is $\beta$ to weight $f_{\text{fine}}$ when combined with $f_{\text{prim}}$ as the final pedestrian representation in testing. We search the two hyper-parameters on a validation set split out from the original training set of Market-1501 (first 651 classes for training and rest 100 classes for validation). Based on the valiation results in Figure 6.8, we choose $\alpha = 0.2$ and $\beta = 0.5$ in all experiments.

**Comparison with state-of-the-art methods.** Finally we report the performance of our approach with other state-of-the-art results in Tables 6.4 and 6.5. Note

that we do not apply any post processing such as re-ranking [191] or multi-query fusion [206]. On each dataset, our approach attains the best performance. Comparing with the methods using separately generated images, DG-Net achieves clear gains of 8.3% and 10.3% for mAP on Market-1501 and DukeMTMC-reID, indicating the advantage of the proposed joint learning. Moreover, our framework is more training efficient: we use only one training phase for joint image generation and re-id learning, while others require two training phases to sequentially train generative models and re-id models. DG-Net also outperforms other non-generative methods by large margins on the two datasets. As for the recent released large-scale dataset MSMT17, DG-Net performs significantly better than the second best method by 9.0% for Rank@1 and 11.9% for mAP.

## 6.4 Summary

In this chapter, we have proposed a joint learning framework that end-to-end couples re-id learning and image generation in a unified network. There exists an online interactive loop between the discriminative and generative modules to mutually benefit the two tasks. Our two modules are co-designed to let the re-id learning better leverage the generated data, rather than simply training on them. Experiments on three benchmarks demonstrate that our approach consistently brings substantial improvements to both image generation quality and re-id accuracy.

Table 6.4 : Comparison with the state-of-the-art methods on the Market-1501 and DukeMTMC-reID datasets. Group 1: the methods not using generated data. Group 2: the methods using separately generated images.

| Methods | Market-1501 | | DukeMTMC-reID | |
|---|---|---|---|---|
| | Rank@1 | mAP | Rank@1 | mAP |
| Verif-Identif [216] | 79.5 | 59.9 | 68.9 | 49.3 |
| DCF [72] | 80.3 | 57.5 | - | - |
| SSM [6] | 82.2 | 68.8 | - | - |
| SVDNet [141] | 82.3 | 62.1 | 76.7 | 56.8 |
| PAN [218] | 82.8 | 63.4 | 71.6 | 51.5 |
| OG-Net-Deep [212] | 86.2 | 68.1 | 76.9 | 57.2 |
| GLAD [172] | 89.9 | 73.9 | - | - |
| HA-CNN [78] | 91.2 | 75.7 | 80.5 | 63.8 |
| MLFN [13] | 90.0 | 74.3 | 81.0 | 62.8 |
| Part-aligned [138] | 91.7 | 79.6 | 84.4 | 69.3 |
| PCB [142] | 93.8 | 81.6 | 83.3 | 69.2 |
| Mancs [161] | 93.1 | 82.3 | 84.9 | 71.8 |
| DeformGAN [134] | 80.6 | 61.3 | - | - |
| LSRO [217] | 84.0 | 66.1 | 67.7 | 47.1 |
| Multi-pseudo [55] | 85.8 | 67.5 | 76.8 | 58.6 |
| PT [87] | 87.7 | 68.9 | 78.5 | 56.9 |
| PN-GAN [118] | 89.4 | 72.6 | 73.6 | 53.2 |
| FD-GAN [33] | 90.5 | 77.7 | 80.0 | 64.5 |
| Ours | **94.8** | **86.0** | **86.6** | **74.8** |

Table 6.5 : Comparison with the state-of-the-art methods on the MSMT17 dataset.

| Methods | Rank@1 | Rank@5 | Rank@10 | mAP |
|---|---|---|---|---|
| Deep [143] | 47.6 | 65.0 | 71.8 | 23.0 |
| OG-Net-Deep [212] | 47.7 | - | - | 23.0 |
| PDC [136] | 58.0 | 73.6 | 79.4 | 29.7 |
| Verif-Identif [216] | 60.5 | 76.2 | 81.6 | 31.6 |
| GLAD [172] | 61.4 | 76.8 | 81.6 | 34.0 |
| PCB [142] | 68.2 | 81.2 | 85.5 | 40.4 |
| Ours | **77.2** | **87.4** | **90.5** | **52.3** |

# Chapter 7

# Conclusions and Future Work

## 7.1 Summary of Contributions

This thesis explored the problem of data limitation and studied the generative and discriminative learning for visual matching. In particular, we

1. proposed a novel semi-supervised learning framework to learn from imperfect generated data for model regularization in Chapter 3;

2. studied the robust learning of visual representation from a new multi-view multi-source dataset including synthetic data simulated by 3D engines in Chapter 4;

3. proposed a two-stage progressive learning strategy to borrow the strength of large-scale real-world data from the web, and demonstrated the scalability of the learned common knowledge in terms of transfer learning in Chapter 5;

4. finally we investigated one unified network for joint generative and discriminative learning, and showed the great benefits of training the generation task with the discriminative task in an end-to-end manner in Chapter 6.

It is clear that the work in this thesis is unable to cover all the potential applications and generalization of representation learning for visual matching. Other directions such as visual feature learning with linguistic descriptions [84, 164, 215, 74], efficient training with millions of data [167, 176, 231], learning from structured information [212, 205] and fast post-processing [201] are also promising directions for

learning robust visual representation to meet the demands of real-world applications. Nevertheless, we believe that our explorations manage to touch the challenging topic of data scarcity in deep learning and are of significant contributions to the field of computer vision in general, paving the way for future studies. Also we believe that the research of joint generative and discriminative learning is just at the beginning and our efforts make learning the knowledge from the freely available data, including generated, synthetic and web data, one step closer to the reliable system for visual matching.

## 7.2 Future Directions

**More Prior Knowledge.** I think the wisdom underpinning the prior knowledge can provide us more insights to future works. Although deeply-learned models outperform many traditional methods [120], especially hand-crafted features, the prediction result is still vulnerable against small visual changes and easy to be cheated [37, 219, 200, 151, 197]. One main reason is that the prior knowledge of humans has not been fully explored in the current prevailing deep learning frameworks [68]. In recent years, more and more researchers have realized this point and try to involve either human-like reasoning [165, 146] or knowledge graph [18] into the current prevailing deep learning frameworks. In terms of visual matching, the common target objects are humans, vehicles and buildings. In this thesis, we study the feasibility of taking advantage of "free" training data to implicitly distill the common visual variants of either the humans, cars or buildings. To take one step further, we observe that humans have one standard 3D geometric structure and explore 3D point cloud of the human body in [212], which explicitly mines the geometric prior knowledge and shows great robustness against occlusion. On the other hand, we also extend the basic geo-localization approach with the spirit of the local binary pattern (LBP) [2] in [166], which enables the contextual information

learning and enrich the representation ability of the learned model on local patterns.

**Efficient Training for More Data.** In this thesis, we explore the potential of involving "free" generated data, synthetic data via 3D engines and web data into the model training process. In the future decades, we may face more and more real-world data, which will be created almost everywhere with mobile devices. But several scientific problems remain: 1. does more data mean higher performance? how to train one high-quality model with millions of data efficiently? 2. how to protect user privacy? For the first question, it may cover several aspects:

1. Data Selection. Some training data is duplicated or similar, which contains limited new information for model to learn. One method is to distill the large dataset to a small one, which has been explored in [167]. But the performance is still far from satisfactory. Another desirable way is to train a life-long learning model, which can preserve the knowledge and is updated only with the forthcoming data [112]. The model trained with large-scale generated data, described in this thesis, can be an appealing starting point for such a learning strategy since we have already let the model "see" many variants.

2. Noisy Annotations. More data generally contains more noisy annotations, which are hard to identify. In Chapter 3, we propose LSRO to provide one smooth label for the imperfect generated data to regularize the model training. Similarly, in Chapter 6, we adopt one teacher model to generate the smooth label for the student model. These two methods prevent the model from over-fitting to the one-hot label, and are robust to the noisy annotation. Recently, we have investigated and applied the data uncertainty to identify the noisy label [214] in the training process. The uncertainty-based method has shown effectiveness in rectifying the noisy annotations, which may become

one alternative choice for future studies.

3. Efficient Model. An efficient model is another key to learning from a large-scale dataset. One favorable solution is to obtain one light-weighted model distilled from a relatively "heavy" model. We have provided one early attempt on pruning a large image retrieval model in [168], which removes the duplicated filter according to clustering results. We think this field has more space to be explored in the future, including mobile CNN models [83, 224] and other variants.

For the second question, as we explored in Chapter 5, transfer learning is one potential solution. In particular, we can train one model with good generalizability on the server then the model is distributed and independently updated on the client terminals. This strategy has been recently adopted in the federated learning [231, 176]. Since federated learning does not need uploading the client data to the server, it can be one desirable way to keep high performance and protect user privacy.

# Bibliography

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *OSDI*, 2016.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *ECCV*, 2004, pp. 469–481.

[3] R. Animus, "Fly high 1 "uiuc" - free creative commons download," https://www.youtube.com/watch?v=jOC-WJW7GAg, August 2015.

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016, pp. 5297–5307.

[5] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *ICML*, 2017.

[6] S. Bai, X. Bai, and Q. Tian, "Scalable person re-identification on supervised smoothed manifold," in *CVPR*, 2017.

[7] Y. Bai, Y. Lou, F. Gao, S. Wang, Y. Wu, and L.-Y. Duan, "Group-sensitive triplet embedding for vehicle reidentification," *TMM*, vol. 20, no. 9, pp. 2385–2399, 2018.

[8] S. Bak, P. Carr, and J.-F. Lalonde, "Domain adaptation through synthesis for unsupervised person re-identification," in *ECCV*, 2018.

[9] I. B. Barbosa, M. Cristani, B. Caputo, A. Rognhaugen, and T. Theoharis, "Looking beyond appearances: Synthetic training data for deep cnns in re-

identification," *Computer Vision and Image Understanding*, vol. 167, pp. 50–62, 2018.

[10] S. Brar, R. Rabbat, V. Raithatha, G. Runcie, and A. Yu, "Drones for deliveries," *Sutardja Center for Entrepreneurship & Technology, University of California, Berkeley, Technical Report*, vol. 8, p. 2015, 2015.

[11] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, "Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss," in *ICCV*, 2019, pp. 8391–8400.

[12] Z. Cai, J. Zhang, D. Ren, C. Yu, H. Zhao, S. Yi, C. K. Yeo, and C. C. Loy, "Messytable: Instance association in multiple camera views," in *ECCV*, 2020.

[13] X. Chang, T. Hospedales, and T. Xiang, "Multi-level factorisation net for person re-identification," in *CVPR*, 2018.

[14] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," *Journal of Machine Learning Research*, vol. 11, no. Mar, pp. 1109–1135, 2010.

[15] B. Chen, W. Deng, and H. Shen, "Virtual class enhanced discriminative embedding learning," in *NeurIPS*, 2018.

[16] D. Chen, Z. Yuan, B. Chen, and N. Zheng, "Similarity learning with spatial constraints for person re-identification," in *CVPR*, 2016.

[17] T.-S. Chen, C.-T. Liu, C.-W. Wu, and S.-Y. Chien, "Orientation-aware vehicle re-identification with semantics-guided part attention network," in *ECCV*, 2020.

[18] W. Chen, W. Xiong, X. Yan, and W. Wang, "Variational knowledge graph reasoning," *arXiv:1803.06581*, 2018.

[19] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *NeurIPS*, 2016.

[20] Y. Chen, X. Zhu, and S. Gong, "Semi-supervised deep learning with memory," in *ECCV*, 2018.

[21] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *CVPR*, 2016.

[22] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, 2018.

[23] R. Chu, Y. Sun, Y. Li, Z. Liu, C. Zhang, and Y. Wei, "Vehicle re-identification with viewpoint-aware metric learning," in *ICCV*, 2019, pp. 8282–8291.

[24] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3893–3903, 2018.

[25] 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.

[26] W. Deng, L. Zheng, G. Kang, Y. Yang, Q. Ye, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person reidentification," in *CVPR*, 2018.

[27] Y. Ding, H. Fan, M. Xu, and Y. Yang, "Adaptive exploration for unsupervised person re-identification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 1, pp. 1–19, 2020.

[28] K. Egan, *Imagination in teaching and learning.* University of Chicago Press, 2014.

[29] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, 1996.

[30] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[31] FlyLow, "Oxford / amazing flight," https://www.youtube.com/watch?v=bs-rwVI_big, May 2016.

[32] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," *arXiv:1706.05208*, 2017.

[33] Y. Ge, Z. Li, H. Zhao, G. Yin, X. Wang, and H. Li, "FD-GAN: Pose-guided feature distilling GAN for robust person re-identification," in *NeurIPS*, 2018.

[34] M. Geng, Y. Wang, T. Xiang, and Y. Tian, "Deep transfer learning for person re-identification," *arXiv:1603.06765*, 2016.

[35] I. Goodfellow, M. Mirza, A. Courville, and Y. Bengio, "Multi-prediction deep boltzmann machines," in *NeurIPS*, 2013.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[37] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *ICLR*, 2015.

[38] H. Guo, C. Zhao, Z. Liu, J. Wang, and H. Lu, "Learning coarse-to-fine structured feature embedding for vehicle re-identification," in *AAAI*, 2018.

[39] B. He, J. Li, Y. Zhao, and Y. Tian, "Part-regularized near-duplicate vehicle re-identification," in *CVPR*, 2019.

[40] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[42] L. He, J. Liang, H. Li, and Z. Sun, "Deep spatial feature reconstruction for partial person re-identification: Alignment-free approach," in *CVPR*, 2018.

[43] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv:1703.07737*, 2017.

[44] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *NeurIPS*, 2017.

[45] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, 2006.

[46] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *ICML*, 2018.

[47] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *ICCV*, 2017.

[48] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.

[49] S. Hu, M. Feng, R. M. Nguyen, and G. Hee Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *CVPR*, 2018, pp. 7258–7267.

[50] S. Hu and X. Chang, "Multi-view drone-based geo-localization via style and spatial alignment," *arXiv:2006.13681*, 2020.

[51] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[52] T.-W. Huang, J. Cai, H. Yang, H.-M. Hsu, and J.-N. Hwang, "Multi-view vehicle re-identification using temporal attention model and metadata re-ranking," in *CVPR Workshops*, 2019.

[53] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization." in *ICCV*, 2017.

[54] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *ECCV*, 2018.

[55] Y. Huang, J. Xu, Q. Wu, Z. Zheng, Z. Zhang, and J. Zhang, "Multi-pseudo regularized label for generated samples in person re-identification," *TIP*, 2018.

[56] Z. Huang, Z. Zheng, C. Yan, H. Xie, Y. Sun, J. Wang, and J. Zhang, "Real-world automatic makeup via identity preservation makeup net," in *IJCAI*, 2020.

[57] O. Ignatova, S. Kalyuga, and J. Sweller, "The imagination effect when using textual or diagrammatic material to learn a second language," *Language Teaching Research*, p. 1362168820971785, 2020.

[58] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[59] X. Jin, C. Lan, W. Zeng, and Z. Chen, "Uncertainty-aware multi-shot knowledge distillation for image-based object re-identification," in *AAII*, 2020.

[60] M. Kalayeh, E. Basaran, M. Gökmen, M. Kamasak, and M. Shah, "Human semantic parsing for person re-identification," in *CVPR*, 2018.

[61] P. Khorramshahi, A. Kumar, N. Peri, S. S. Rambhatla, J.-C. Chen, and R. Chellappa, "A dual path modelwith adaptive attention for vehicle re-identification," in *ICCV*, 2019.

[62] P. Khorramshahi, N. Peri, J.-c. Chen, and R. Chellappa, "The devil is in the details: Self-supervised attention for vehicle re-identification," in *ECCV*, 2020.

[63] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[64] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof, "Large scale metric learning from equivalence constraints," in *CVPR*, 2012.

[65] J. Krause, B. Sapp, A. Howard, H. Zhou, A. Toshev, T. Duerig, J. Philbin, and L. Fei-Fei, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *ECCV*, 2016.

[66] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *3DRR*, 2013.

[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[68] K. Kuang, L. Li, Z. Geng, L. Xu, K. Zhang, B. Liao, H. Huang, P. Ding, W. Miao, and Z. Jiang, "Causal inference," *Engineering*, vol. 6, no. 3, pp. 253–263, 2020.

[69] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *ICLR*, 2016.

[70] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, 2013.

[71] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *ECCV*, 2018.

[72] D. Li, X. Chen, Z. Zhang, and K. Huang, "Learning deep context-aware features over body and latent parts for person re-identification," in *CVPR*, 2017.

[73] P. Li, Y. Wei, and Y. Yang, "Meta parsing networks: Towards generalized few-shot scene parsing with adaptive metric learning," in *ACM Multimedia*, 2020.

[74] S. Li, T. Xiao, H. Li, B. Zhou, D. Yue, and X. Wang, "Person search with natural language description," in *CVPR*, 2017, pp. 1970–1979.

[75] S. Li and D.-Y. Yeung, "Visual object tracking for unmanned aerial vehicles: A benchmark and new motion models," in *AAAI*, 2017.

[76] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014.

[77] W. Li, X. Zhu, and S. Gong, "Person re-identification by deep joint learning of multi-loss classification," in *IJCAI*, 2017.

[78] ——, "Harmonious attention network for person re-identification," in *CVPR*, 2018.

[79] X. Li, A. Wu, and W.-S. Zheng, "Adversarial open-world person re-identification," in *ECCV*, 2018.

[80] S. Liao, Y. Hu, X. Zhu, and S. Li, "Person re-identification by local maximal occurrence representation and metric learning," in *CVPR*, 2015.

[81] K. Lin, L. Wang, K. Luo, Y. Chen, Z. Liu, and M.-T. Sun, "Cross-domain complementary learning using pose for multi-person part segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[82] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *CVPR*, 2015.

[83] X. Lin, C. Zhao, and W. Pan, "Towards accurate binary convolutional neural network," *arXiv:1711.11294*, 2017.

[84] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, Z. Hu, C. Yan, and Y. Yang, "Improving person re-identification by attribute and identity learning," *Pattern Recognition*, vol. 95, pp. 151–161, 2019.

[85] Y. Lin, Z. Zheng, H. Zhang, C. Gao, and Y. Yang, "Bayesian query expansion for multi-camera person re-identification," *Pattern Recognition Letters*, 2018.

[86] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, "Deep relative distance learning: Tell the difference between similar vehicles," in *CVPR*, 2016.

[87] J. Liu, B. Ni, Y. Yan, P. Zhou, S. Cheng, and J. Hu, "Pose transferrable person re-identification," in *CVPR*, 2018.

[88] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," *CVPR*, 2019.

[89] L. Liu, H. Li, and Y. Dai, "Stochastic attraction-repulsion embedding for large scale image localization," in *ICCV*, 2019, pp. 2570–2579.

[90] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, "Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set," in *CVPR*, 2012.

[91] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *ICML*, 2016.

[92] X. Liu, J. Wang, S. Wen, E. Ding, and Y. Lin, "Localizing by describing: Attribute-guided attention localization for fine-grained recognition," in *AAAI*, 2017.

[93] X. Liu, S. Zhang, X. Wang, R. Hong, and Q. Tian, "Group-group loss-based global-regional feature learning for vehicle re-identification," *IEEE Transactions on Image Processing*, vol. 29, pp. 2638–2652, 2019.

[94] X.-C. Liu, H.-D. Ma, and S.-Q. Li, "Pvss: A progressive vehicle search system for video surveillance networks," *Journal of Computer Science and Technology*, vol. 34, no. 3, pp. 634–644, 2019.

[95] X. Liu, W. Liu, T. Mei, and H. Ma, "A deep learning-based approach to progressive vehicle re-identification for urban surveillance," in *ECCV*, 2016.

[96] ——, "Provid: Progressive and multimodal vehicle reidentification for large-scale urban surveillance," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 645–658, 2017.

[97] X. Liu, W. Liu, J. Zheng, C. Yan, and T. Mei, "Beyond the parts: Learning multi-view cross-part correlation for vehicle re-identification," in *ACM Multimedia*, 2020.

[98] Y. Lou, Y. Bai, J. Liu, S. Wang, and L. Duan, "Veri-wild: A large dataset and a new method for vehicle re-identification in the wild," in *CVPR*, 2019.

[99] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *CVPR*, 2019.

[100] Y. Luo, Z. Zheng, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Macro-micro adversarial network for human parsing," in *ECCV*, 2018.

[101] K. Lv, W. Deng, Y. Hou, H. Du, H. Sheng, J. Jiao, and L. Zheng, "Vehicle reidentification with the location and time stamp," in *CVPR Workshops*, 2019.

[102] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *NeurIPS*, 2017.

[103] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, "Exploring the limits of weakly supervised pretraining," in *ECCV*, 2018.

[104] X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, and S. Smolley, "Least squares generative adversarial networks," in *ICCV*, 2017.

[105] D. Meng, L. Li, X. Liu, Y. Li, S. Yang, Z.-J. Zha, X. Gao, S. Wang, and Q. Huang, "Parsing-based view-aware embedding network for vehicle reidentification," in *CVPR*, 2020.

[106] J. Meng, S. Wu, and W.-S. Zheng, "Weakly supervised person reidentification," in *CVPR*, 2019, pp. 760–769.

[107] L. Mescheder, S. Nowozin, and A. Geiger, "Which training methods for GANs do actually converge?" in *ICML*, 2018.

[108] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv:1606.01583*, 2016.

[109] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *CVPR*, 2016, pp. 4004–4012.

[110] S. J. Pan and Q. Yang, "A survey on transfer learning," *TKDE*, vol. 22, no. 10, pp. 1345–1359, 2009.

[111] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille, "Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation," in *ICCV*, 2015.

[112] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[113] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.

[114] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *CVPR*, 2007.

[115] ——, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *CVPR*, 2008.

[116] J. Qian, W. Jiang, H. Luo, and H. Yu, "Stripe-based and attribute-aware network: A two-branch deep model for vehicle re-identification," *Measurement Science and Technology*, vol. 31, no. 9, p. 095401, 2020.

[117] X. Qian, Y. Fu, Y.-G. Jiang, T. Xiang, and X. Xue, "Multi-scale deep learning architectures for person re-identification," in *CVPR*, 2017.

[118] X. Qian, Y. Fu, T. Xiang, W. Wang, J. Qiu, Y. Wu, Y.-G. Jiang, and X. Xue, "Pose-normalized image generation for person re-identification," in *ECCV*, 2018.

[119] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *CVPR*, 2018.

[120] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," *IEEE transactions on pattern analysis and machine*

*intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[121] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ICLR*, 2016.

[122] M. Ranzato and M. Szummer, "Semi-supervised learning of compact document representations with deep networks," in *ICML*, 2008.

[123] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *NeurIPS*, 2015.

[124] K. Regmi and M. Shah, "Bridging the domain gap for ground-to-aerial image matching," in *ICCV*, 2019, pp. 470–479.

[125] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *ECCV*, ser. LNCS, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906. Springer International Publishing, 2016, pp. 102–118.

[126] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCVW*, 2016.

[127] E. Ristani and C. Tomasi, "Features for multi-target multi-camera tracking and re-identification," in *CVPR*, 2018.

[128] T. A. Rule, "Airspace in an age of drones," *BUL Rev.*, vol. 95, p. 155, 2015.

[129] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[130] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *NeurIPS*, 2016.

[131] Y. Shen, T. Xiao, H. Li, S. Yi, and X. Wang, "Learning deep neural networks for vehicle re-id with visual-spatio-temporal path proposals," in *ICCV*, 2017.

[132] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," in *NeurIPS*, 2019, pp. 10 090–10 100.

[133] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," *AAAI*, 2020.

[134] A. Siarohin, E. Sangineto, S. Lathuilière, and N. Sebe, "Deformable GANs for pose-based human image generation," in *CVPR*, 2018.

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

[136] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Pose-driven deep convolutional model for person re-identification," in *ICCV*, 2017.

[137] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Deep attributes driven multi-camera person re-identification," in *ECCV*, 2016.

[138] Y. Suh, J. Wang, S. Tang, T. Mei, and K. M. Lee, "Part-aligned bilinear representations for person re-identification," in *ECCV*, 2018.

[139] X. Sun and L. Zheng, "Dissecting person re-identification from the viewpoint of viewpoint," in *CVPR*, 2019.

[140] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *CVPR*, 2020.

[141] Y. Sun, L. Zheng, W. Deng, and S. Wang, "SVDNet for pedestrian retrieval," in *ICCV*, 2017.

[142] Y. Sun, L. Zheng, Y. Yang, Q. Tian, and S. Wang, "Beyond part models: Person retrieval with refined part pooling," in *ECCV*, 2018.

[143] 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*, 2015.

[144] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016.

[145] X. Tan, Z. Wang, M. Jiang, X. Yang, J. Wang, Y. Gao, X. Su, X. Ye, Y. Yuan, D. He *et al.*, "Multi-camera vehicle tracking and re-identification based on visual and spatial-temporal features," in *CVPR Workshops*, 2019.

[146] K. Tang, J. Huang, and H. Zhang, "Long-tailed classification by keeping the good and removing the bad momentum causal effect," in *NeurIPS*, 2020.

[147] Z. Tang, M. Naphade, S. Birchfield, J. Tremblay, W. Hodge, R. Kumar, S. Wang, and X. Yang, "Pamtri: Pose-aware multi-task learning for vehicle re-identification using highly randomized synthetic data," in *ICCV*, 2019.

[148] Z. Tang, M. Naphade, M.-Y. Liu, X. Yang, S. Birchfield, S. Wang, R. Kumar, D. Anastasiu, and J.-N. Hwang, "Cityflow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification," in *CVPR*, 2019.

[149] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *NeurIPS*, 2017.

[150] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," in *CVPR*, 2017.

[151] G. Tolias, F. Radenovic, and O. Chum, "Targeted mismatch adversarial attack: Query with a flower to retrieve the tower," in *ICCV*, 2019, pp. 5037–5046.

[152] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *CVPR*, 2015.

[153] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *CVPR*, 2018.

[154] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker, "Domain adaptation for structured output via discriminative patch representations," in *ICCV*, 2019.

[155] E. Ustinova, Y. Ganin, and V. Lempitsky, "Multiregion bilinear convolutional neural networks for person re-identification," *arXiv:1512.05300*, 2015.

[156] R. R. Varior, M. Haloi, and G. Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *ECCV*, 2016.

[157] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang, "A siamese long short-term memory architecture for human re-identification," in *ECCV*, 2016.

[158] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *ACM Multimedia*, 2015.

[159] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *ECCV*, 2016.

[160] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[161] C. Wang, Q. Zhang, C. Huang, W. Liu, and X. Wang, "Mancs: A multi-task attentional network with curriculum sampling for person re-identification," in *ECCV*, 2018.

[162] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang, "Multiple granularity descriptors for fine-grained categorization," in *ICCV*, 2015.

[163] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *CVPR*, 2016.

[164] J. Wang, X. Zhu, S. Gong, and W. Li, "Transferable joint attribute-identity deep learning for unsupervised person re-identification," in *CVPR*, 2018.

[165] T. Wang, J. Huang, H. Zhang, and Q. Sun, "Visual commonsense representation learning via causal inference," in *CVPR Workshops*, 2020.

[166] T. Wang, Z. Zheng, C. Yan, and Y. Yang, "Each part matters: Local patterns facilitate cross-view geo-localization," *TCSVT*, 2021.

[167] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *arXiv:1811.10959*, 2018.

[168] X. Wang, Z. Zheng, Y. He, F. Yan, Z. Zeng, and Y. Yang, "Progressive local filter pruning for image retrieval acceleration," *arXiv:2001.08878*, 2020.

[169] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang, "Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification," in *ICCV*, 2017.

[170] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *TIP*, 2004.

[171] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer GAN to bridge domain gap for person re-identification," in *CVPR*, 2018.

[172] L. Wei, S. Zhang, H. Yao, W. Gao, and Q. Tian, "Glad: global-local-alignment descriptor for pedestrian retrieval," in *ACM Multimedia*, 2017.

[173] J. Weston, S. Chopra, and A. Bordes, "Memory networks," *arXiv:1410.3916*, 2014.

[174] S. Workman and N. Jacobs, "On the location dependence of convolutional neural network features," in *CVPR Workshops*, 2015, pp. 70–78.

[175] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *ICCV*, 2015, pp. 3961–3969.

[176] G. Wu and S. Gong, "Decentralised learning from independent multi-domain labels for person re-identification," *arXiv:2006.04150*, 2020.

[177] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," in *NeurIPS*, 2016.

[178] L. Wu, C. Shen, and A. v. d. Hengel, "Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification," *Pattern Recognition*, 2016.

[179] L. Wu, Y. Wang, J. Gao, and X. Li, "Where-and-when to look: Deep siamese attention networks for video-based person re-identification," *TMM*, 2018.

[180] Y. Wu, Y. Lin, X. Dong, Y. Yan, W. Bian, and Y. Yang, "Progressive learning for person re-identification with one example," *TIP*, 2019.

[181] Z. Wu, X. Han, Y.-L. Lin, M. Gokhan Uzunbas, T. Goldstein, S. Nam Lim, and L. S. Davis, "Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation," in *ECCV*, 2018.

[182] Z. Wu, X. Wang, J. E. Gonzalez, T. Goldstein, and L. S. Davis, "Ace: Adapting to changing environments for semantic segmentation," in *ICCV*, 2019.

[183] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, "Joint detection and identification feature learning for person search," in *CVPR*, 2017, pp. 3415–3424.

[184] K. Yan, Y. Tian, Y. Wang, W. Zeng, and T. Huang, "Exploiting multi-grain ranking constraints for precisely searching visually-similar vehicles," in *ICCV*, 2017.

[185] L. Yang, P. Luo, C. Change Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *CVPR*, 2015.

[186] Y. Yang, D. Xu, F. Nie, J. Luo, and Y. Zhuang, "Ranking with local regression and global alignment for cross media retrieval," in *ACM Multimedia*, 2009, pp. 175–184.

[187] Y. Yao, L. Zheng, X. Yang, M. Naphade, and T. Gedeon, "Simulating content consistent vehicle datasets with attribute descent," *ECCV*, 2020.

[188] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *CVPR*, 2017.

[189] D. Yi, Z. Lei, S. Liao, and S. Li, "Deep metric learning for person re-identification," in *ICPR*, 2014.

[190] Q. Yu, C. Wang, B. Cetiner, S. X. Yu, F. Mckenna, E. Taciroglu, and K. H. Law, "Building information modeling and classification by visual learning at

a city scale," *NeurIPS Workshop*, 2019.

[191] R. Yu, Z. Zhou, S. Bai, and X. Bai, "Divide and fuse: A re-ranking approach for person re-identification," in *BMVC*, 2017.

[192] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in *ICCV*, 2019.

[193] D. Zapletal and A. Herout, "Vehicle re-identification for automatic video traffic surveillance," in *CVPR Workshops*, 2016.

[194] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, "Predicting ground-level scene layout from aerial imagery," in *CVPR*, 2017.

[195] D. Zhang, J. Han, L. Yang, and D. Xu, "Spftn: a joint learning framework for localizing and segmenting objects in weakly labeled videos," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[196] D. Zhang, J. Han, L. Zhao, and D. Meng, "Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework," *International Journal of Computer Vision*, vol. 127, no. 4, pp. 363–380, 2019.

[197] H. Zhang, L. Zhu, Y. Zhu, and Y. Yang, "Motion-excited sampler: Video adversarial attack with sparked prior," in *ECCV*, 2020.

[198] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," *CVPR*, 2016.

[199] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *ECCV*, 2014.

[200] R. Zhang, "Making convolutional networks shift-invariant again," in *ICML*. PMLR, 2019, pp. 7324–7334.

[201] X. Zhang, M. Jiang, Z. Zheng, X. Tan, E. Ding, and Y. Yang, "Understanding image retrieval re-ranking: A graph neural network perspective," *arXiv:2012.07620*, 2020.

[202] Y. Zhang, D. Liu, and Z.-J. Zha, "Improving triplet-wise training of convolutional neural network for vehicle re-identification," in *ICME*, 2017.

[203] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *CVPR*, 2018.

[204] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang, "Spindle net: Person re-identification with human body region guided feature decomposition and fusion," in *CVPR*, 2017.

[205] L. Zheng, Y. Huang, H. Lu, and Y. Yang, "Pose invariant embedding for deep person re-identification," *arXiv:1701.07732*, 2017.

[206] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *ICCV*, 2015.

[207] L. Zheng, Y. Yang, and A. Hauptmann, "Person re-identification: Past, present and future," *arXiv:1610.02984*, 2016.

[208] L. Zheng, H. Zhang, S. Sun, M. Chandraker, and Q. Tian, "Person re-identification in the wild," *CVPR*, 2017.

[209] Z. Zheng, T. Ruan, Y. Wei, Y. Yang, and T. Mei, "Vehiclenet: learning robust visual representation for vehicle re-identification," *IEEE Transactions on Multimedia*, 2020.

[210] Z. Zheng, Y. Wei, and Y. Yang, "University-1652: A multi-view multi-source benchmark for drone-based geo-localization," *ACM Multimedia*, 2020.

[211] Z. Zheng, X. Yang, Z. Yu, L. Zheng, Y. Yang, and J. Kautz, "Joint discriminative and generative learning for person re-identification," *CVPR*, 2019.

[212] Z. Zheng and Y. Yang, "Person re-identification in the 3d space," *arXiv:2006.04569*, 2020.

[213] ——, "Unsupervised scene adaptation with memory regularization in vivo," *IJCAI*, 2020.

[214] ——, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *International Journal of Computer Vision*, pp. 1–15, 2021.

[215] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu, and Y.-D. Shen, "Dual-path convolutional image-text embeddings with instance loss," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 2, pp. 1–23, 2020.

[216] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned cnn embedding for person reidentification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–20, 2017.

[217] ——, "Unlabeled samples generated by GAN improve the person re-identification baseline in vitro," in *ICCV*, 2017.

[218] ——, "Pedestrian alignment network for large-scale person re-identification," *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[219] Z. Zheng, L. Zheng, Y. Yang, and F. Wu, "Query attack via opposite-direction feature: Towards robust image retrieval," *arXiv preprint arXiv:1809.02681*,

2018.

[220] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *CVPR*, 2017.

[221] Z. Zhong, L. Zheng, S. Li, and Y. Yang, "Generalizing a person retrieval model hetero-and homogeneously," in *ECCV*, 2018.

[222] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Invariance matters: Exemplar memory for domain adaptive person re-identification," in *CVPR*, 2019.

[223] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[224] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Omni-scale feature learning for person re-identification," in *ICCV*, 2019.

[225] Y. Zhou and L. Shao, "Cross-view gan based vehicle generation for re-identification." in *BMVC*, vol. 1, 2017, pp. 1–12.

[226] ——, "Aware attentive multi-view inference for vehicle re-identification," in *CVPR*, 2018.

[227] J. Zhu, H. Zeng, J. Huang, S. Liao, Z. Lei, C. Cai, and L. Zheng, "Vehicle re-identification using quadruple directional deep learning features," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[228] J.-Y. Zhu, T. Park, P. Isola, and A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *ICCV*, 2017.

[229] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman, "Toward multimodal image-to-image translation," in *NeurIPS*, 2017.

[230] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," *arXiv:1804.07437*, 2018.

[231] W. Zhuang, Y. Wen, X. Zhang, X. Gan, D. Yin, D. Zhou, S. Zhang, and S. Yi, "Performance optimization of federated person re-identification via benchmark analysis," in *ACM Multimedia*, 2020.

[232] Y. Zou, Z. Yu, V. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *ECCV*, 2018.

[233] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *ICCV*, 2019.