Scalable Deep Learning Architecture Design

Wei Li

Submitted in partial fulfilment of the requirement for the degree of *Doctor of Philosophy*

School of Electronic Engineering and Computer Science

Queen Mary University of London

31 January 2021

Scalable Deep Learning Architecture Design

Wei Li

Abstract

The past decade has witnessed a rapid development in deep learning research which has enabled remarkable progress on a wide spectrum of computer vision tasks, such as object recognition, segmentation, and detection. One generic mechanism for deep learning on computer vision is to *design optimal deep neural architectures for given tasks*, so as to learn compact, rich and expressive features for data collected by artificial visual sensors. Nonetheless, deep artificial neural architecture design for computer vision tasks remains challenging due to the inherent visual task complexity and uncertainty. One can not guarantee that a specific network designed for one task assumably works well for new tasks, especially when it comes to considering scalability (the model size, learning capacity and efficiency, and domain adaptation to new data). Unfortunately, there are no theoretical principles towards guiding deep neural architecture design, which makes researchers having to rely on their own expertise and experience ad hoc. This thesis investigates approaches to designing deep neural architectures for several tasks by considering the underlying task characteristics for more efficient and powerful deep models. More specifically, this thesis develops new methods for addressing four different problems as follows:

Chapter 3 The first problem is *harmonious attention network design for scalable person re-identification (re-id).* Existing person re-identification (re-id) deep learning methods rely heavily on the utilisation of large and computationally expensive convolutional neural networks. They are therefore *not scalable* to large scale re-id deployment scenarios with the need of processing a large amount of surveillance video data, due to the lengthy inference process with high computing costs. in this chapter, we address this limitation via jointly learning re-id attention selection. Specifically, we formulate a novel Harmonious Attention Network (HAN) framework to jointly learn soft pixel attention and hard regional attention alongside simultaneous deep feature representation learning, particularly enabling more discriminative re-id matching by *efficient* networks with more scalable inference. Extensive evaluations validate the cost-effectiveness superiority of the proposed HAN approach for person re-id against a wide variety of state-of-the-art methods on large benchmark datasets.

Chapter 4 The second problem is *hierarchical distillation network design for scalable person search.* Existing person search methods typically focus on improving person detection accuracy. This ignores the model inference efficiency, which however is fundamentally significant for real-world applications. in this chapter, we address this limitation by investigating the scalability problem of person search involving both model accuracy and inference efficiency simultaneously. Specifically, we formulate a Hierarchical Distillation Learning (HDL) approach. With HDL, we aim to comprehensively distil the knowledge of a strong teacher model with strong learning capability to a lightweight student model with weak learning capability. To facilitate the HDL process, we design a simple and powerful teacher model for joint learning of person detection and person re-identification matching in unconstrained scene images. Extensive experiments show the modelling advantages and cost-effectiveness superiority of HDL over the

state-of-the-art person search methods on large person search benchmarks.

Chapter 5 The third problem is *neural graph embedding for scalable neural architecture search.* Existing neural architecture search (NAS) methods often operate in discrete or continuous spaces *directly*, which ignores the *graphical topology knowledge* of neural networks. This leads to suboptimal search performance and efficiency, given that neural networks are essentially *directed acyclic graphs* (DAG). in this chapter, we address this limitation by introducing a novel idea of *neural graph embedding* (NGE). Specifically, we represent the building block (i.e. the cell) of neural networks with a neural DAG, and learn it by leveraging a Graph Convolutional Network to propagate and model the intrinsic topology information of network architectures. This results in a generic neural network representation integrable with different existing NAS frameworks. Extensive experiments show the superiority of NGE over the state-of-the-art methods on image classification and semantic segmentation.

Chapter 6 The last problem is *scalable neural operator search.* Existing neural architecture search (NAS) methods explore a limited *feature-transformation-only* search space, ignoring other advanced feature operations such as feature self-calibration by attention and dynamic convolutions. This disables the NAS algorithms from discovering more optimal network architectures. We address this limitation by additionally exploiting feature self-calibration operations, resulting in a heterogeneous search space. To overcome the challenges of operation heterogeneity and significantly larger search space, we formulate a *neural operator search* (NOS) method. NOS presents a novel heterogeneous residual block for integrating the heterogeneous operations in a unified structure, and an attention guided search strategy for facilitating the search process over a vast space. Extensive experiments show that NOS can search novel cell architectures with highly competitive performance on the CIFAR and ImageNet benchmarks.

Chapter 6 includes concluding remarks and discusses potential areas for future research and extensions.

Declaration

I hereby declare that this thesis has been composed by myself and that it describes my own work. It has not been submitted, either in the same or different form, to this or any other university for a degree. All verbatim extracts are distinguished by quotation marks, and all sources of information have been acknowledged.

Some parts of the work have previously been published or in submission as:

Chapter 3

- W. Li, X. Zhu, S. Gong, *Person Re-Identification by Deep Joint Learning of Multi-Loss Classification*, The Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI), 2017.

- W. Li, X. Zhu, S. Gong, *Harmonious Attention Network for Person Re-Identification*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

- W. Li, X. Zhu, S. Gong, *Scalable Person Re-Identification by Harmonious Attention*, International Journal of Computer Vision (IJCV), 2019.

Chapter 4

- W. Li, S. Gong, X. Zhu, *Hierarchical Distillation Learning for Scalable Person Search*, submitted to Pattern Recognition (PR), 2020.

Chapter 5

- W. Li, S. Gong, X. Zhu, *Neural Graph Embedding for Neural Architecture Search*, The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI), 2020.

Chapter 6

- W. Li, S. Gong, X. Zhu,, *Neural Operator Search*, submitted to The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI), 2021.

Acknowledgements

Foremost, I would like to express my greatest gratitude to my supervisor, Prof. Shaogang (Sean) Gong, for his consistent support and encouragement, as well as all those bright ideas and wonderful opportunities he has kindly provided to me. When I was stuck in failures, Sean always gives plenty of valuable advice and unconditional supports. I would like to express my deep gratefulness to Dr. Xiatian Zhu for his helpful discussions and encouragement. Without their outstanding support throughout my PhD study, I would never became an independent researcher.

My warm appreciation goes to everyone I met at the Vision Group for their friendship and support: Qian Yu, Feng Liu, Li Zhang, Elyor Kodirov, Jingya Wang, Ying Zhang, Qi Dong, Xiaobin Chang, Jifei Song, Hang Su, Da Li, Xu Lan, Tianyuan Yu, Zhiyi Cheng, Kaiyue Pang, Ke Li, Anran Qi, Conghui Hu, Yanbei Chen, Umar Riaz Muhammad, Jiabo Huang, Minxian Li, Guile Wu, Aytac Kanaci. I am very grateful to the EECS system supports and administrative staff, especially Tim Kay for solving computing server issues in all my research projects and Melissa Yeo for taking care of my progresses in PhD study.

My very deep and sincere gratitude to my family for their continuous and selflessness love. This journey would not have been possible without their endless support.

Contents

1 Introduction **17**

1.1 Deep Learning Architecture Design in Computer Vision 17

1.1.1 Definition of Deep Learning Architecture Design 19

1.1.2 Challenges in Scalable Deep Learning Architecture Design 21

1.2 Contributions 22

1.3 Thesis Outline 24

2 Literature Review **27**

2.1 Handcrafted or Manually-Designed Deep Architectures 27

2.2 Neural Architecture Search (NAS) 28

2.3 The Person Re-Identification Problem 32

2.4 The Person Search Problem 35

2.5 Knowledge Distillation 38

3 Scalable Attention Network Design **41**

3.1 Overview 41

3.2 Scalable Person Re-Identification 42

3.2.1 Global-Local Feature Learning 44

3.2.2 Harmonious Attention Learning 46

3.2.3 HAN Model Instantiation 51

3.2.4 Scalable Person Re-ID by HAN 52

3.3 Experiments 53

3.3.1 Further Analysis and Discussions 54

3.3.2 Comparing State-of-the-Art Re-ID Methods 59

3.3.3 Comparing State-of-the-Art Efficient Networks 64

3.4 Summary 65

4 Scalable Distillation Network Design 67

4.1 Overview 67

4.2 Hierarchical Distillation Learning 67

4.2.1 A Strong Joint Learning Teacher Model 68

4.2.2 An Efficient Joint Learning Student Model 71

4.2.3 Hierarchical Distillation Learning 71

4.2.4 Model Training 74

4.2.5 Network Architecture Details 75

4.3 Experiments 75

4.3.1 Further Analysis and Discussions 77

4.3.2 Comparisons to State-of-the-Art Methods 78

4.4 Summary 80

5 Scalable Neural Architecture Search 81

5.1 Overview 81

5.2 NAS as Optimisation 81

5.3 Methodology 82

5.3.1 A Neural Graph 83

5.3.2 NGE for NAS 85

5.4 Experiments 86

5.4.1 Datasets 86

5.4.2 NGE Instantiating 87

5.4.3 Cell Search 88

5.4.4 Further Analysis 88

5.4.5 Architecture Evaluation 89

5.5 Summary 92

6 Scalable Neural Operator Search 93

6.1 Overview 93

6.2 Method 94

6.2.1 Heterogeneous Search Space 94

6.2.2 Heterogeneous Operator Cell 97

6.2.3 Attention Guided Search Optimisation in a Heterogeneous Search Space . 98
6.3 Experiments 100
6.3.1 Details of Training Configurations 101
6.3.2 Study of Feature Self-Calibration Operations 102
6.3.3 Cell Search 103
6.3.4 Further Analysis 104
6.3.5 Architecture Evaluation 105
6.4 Summary 106

7 Conclusion and Future Work **107**
7.1 Conclusion 107
7.2 Future Work 109

Bibliography **111**

List of Figures

1.1 Deep learning for Visual Recognition Tasks 17

1.2 The statistics of performance in Large Scale Visual Recognition Challenges . . . 18

1.3 An abstract illustration of deep neural architecture design 19

1.4 An abstract illustration of challenges in deep neural architecture design 21

2.1 Some examples of handcrafted or manually-designed deep architectures 28

2.2 Overview of building the network 31

2.3 Illustration of person re-identification person 32

2.4 The pipeline of a person re-identification system. 32

2.5 The significance of scalability in person search 36

2.6 A Generic Teacher-student Framework for Knowledge Distillation. 38

3.1 Schematic architecture of the proposed Harmonious Attention Network (HAN) . 43

3.2 Structure of a Harmonious Attention module 47

3.3 Schematic comparison between Spatial Transformer Network (STN) and HAN Hard Attention 50

3.4 Structure of local block and global block. 52

3.5 Example cross-view matched person image pairs 53

3.6 Evaluating a set of feature dimensions 56

3.7 Visualisation of the harmonious re-id attention 58

4.1 Overview of the proposed Hierarchical Distillation Learning (HDL) approach . . 68

4.2 A student's building block contains three module 71

4.3 Attention residual module in knowledge distillation. 72

4.4 Example query and unconstrained scene images 75

4.5 Test mAP of varying gallery sizes on CUHK-SYSU. 79

5.1 The concept of neural graph architecture search 82

5.2 An overview of the proposed Neural Graph Embedding (NGE) for NAS 83

5.3 Normal cell and reduction cell obtained by NGE. 87

6.1 Structure of the proposed dynamic convolutions for image classification 94

6.2 Heterogeneous Residual block for formulating the inner node computation 96

6.3 Overview of attention guided search 96

6.4 Visualisation of search space 100

6.5 Normal cell and reduction cell searched on CIFAR-10 102

6.6 The train and val set accuracies on CIFAR10 105

List of Acronyms

Re-Id	Re-Identification
NAS	Neural Architecture Search
NOS	Neural Operator Search
CNN	Convolutional Neural Network
CMC	Cumulative Match Characteristic
mAP	Mean Average Precision
R1	Rank-1 Recognition Rage
SGD	Stochastic Gradient Descent

Chapter 1

Introduction

Scientists in the computer vision community have long dreamed of developing computers that have the like of the human eyes' abilities to interpret the surrounding environment. Towards this goal, a fundamental task is to extract compact, rich and expressive features for data collected by artificial visual sensors. Over the past decade, this desire has been advanced notably by the breakthrough in deep learning, making significant progress on a wide spectrum of computer vision tasks, such as object recognition, segmentation, and detection. It leads to a major shift of research focus in computer vision from conventional hand-crafted feature engineering to deep representation learning. Apart from the critical process of model training, as shown in Figure 1.1, one fundamental procedure is to design appropriate deep neural architectures to learn optimal representations for different computer vision tasks.

1.1 Deep Learning Architecture Design in Computer Vision

For most of current computer vision research, the deep neural network has become a fundamental functionality which exacts expressive representations for numerous high level and complex

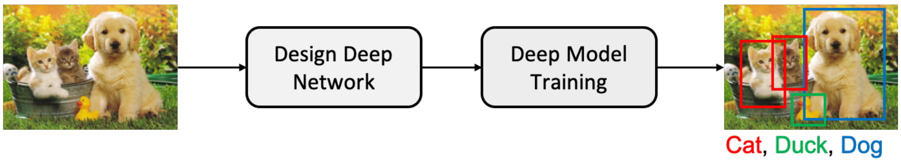


Figure 1.1: Deep learning for Visual Recognition Tasks.

visual applications. Among the various applications that deep neural networks can bring improvements, the most significant application is the large scale image recognition. Thanks to the hardware innovations and the availability of cheaper and highly efficient computational devices (*e.g.* GPUs), the performance in Large Scale Visual Recognition Challenge (ILSVRC) [1] every year has been fast-climbing since the birth of the groundbreaking AlexNet [1] (see Figure 1.2). One might be impressed by the human-level recognition performance achieved by a deep neural network on ImageNet [2] benchmark. However, designing a specific deep neural network that could perform well on a given task is non-trivial that normally needs a lot of trial and error, with high demand for a researcher's expertise and experience. Substantial efforts have been made towards finding good practices for designing state-of-the-art deep neural networks. Taking the ResNet [3] as an example, it is an enormous architecture with skip connections all over, which is the winner of ILSVRC 2015 and MS COCO 2015 [2] in image classification, detection, and segmentation. The success of ResNet was achieved by standing on the shoulders of pioneers (*e.g.* AlexNet [1], VGGNet [4], and GoogleNet [5]). As shown in Figure 1.2, it took almost half a decade (2010-2015) to realise this evolution from the seminal AlexNet to the exceptional ResNet that greatly surpassed human-level performance on ImageNet classification.

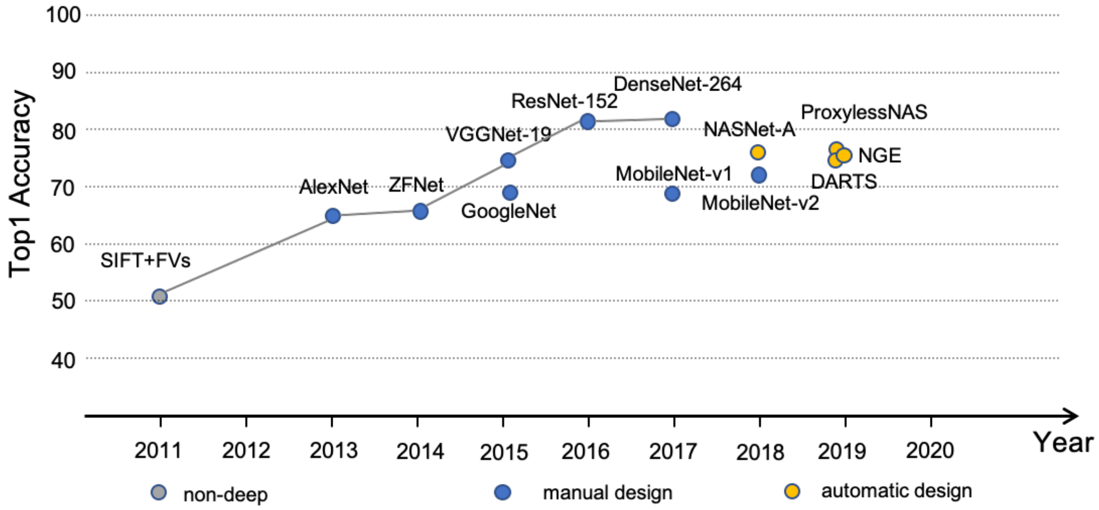


Figure 1.2: The statistics of performance in Large Scale Visual Recognition Challenges.

In order to alleviate the demands for human knowledge and interventions, recently there is an ongoing research trend—Neural Architecture Search (NAS)—that aims to automate the tedious process of designing neural network architectures optimal for target tasks. By far, recent attempts

[1] http://www.image-net.org/

[2] http://image-net.org/challenges/ilsvrc+mscoco2015

in NAS have achieved enormous success in various challenging tasks, *e.g*. image classification [6], object detection [7], and semantic segmentation [8, 9]. Existing NAS methods usually fall into three categories: reinforcement learning (RL) based methods, evolutionary algorithm (EA) based methods, and gradient differentiable (GD) methods. For example, the policy networks in [10, 11] guide the selection of the architecture component sequentially. Some EA-based methods [12, 13] evolve a population of initialised architectures with the corresponding validation accuracies as fitness. Instead of searching in a discrete search space, DARTS [14] provides a gradient optimisation NAS framework, in which the search space is relaxed to be continuous. Several works [15, 12] attempt to reduce the search cost by exploring the search space progressively.

1.1.1 Definition of Deep Learning Architecture Design

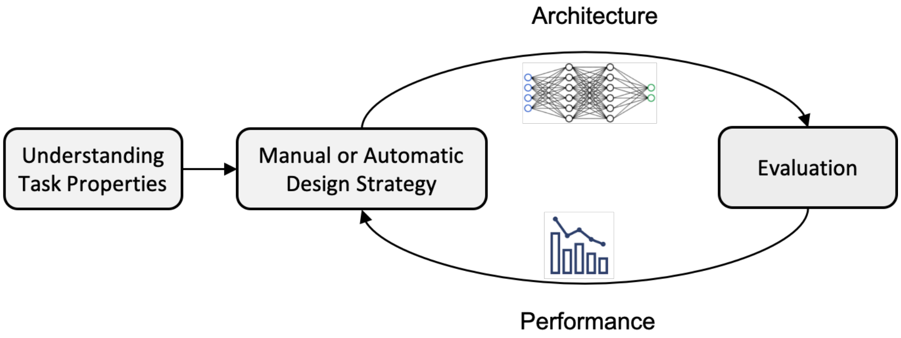

Architecture
Understanding Task Properties
Manual or Automatic Design Strategy
Evaluation
Performance

Figure 1.3: An abstract illustration of deep neural architecture design.

Due to lacking fundamental theoretical principles, researchers have to rely on their expertise and experience ad hoc to design a proper architecture through trial and error. Abstractly, the process of designing a deep neural architecture for given tasks contains three main steps: a) understanding task properties; b) manual or automatic design strategy; c) performance evaluation. As conceptually illustrated in Figure 1.3, this is not a one-way procedure but requires multiple iterations of refinement and optimisation. Specifically, a manual or automatic strategy proposes an architecture by understanding task properties. The architecture is passed to a performance evaluation step, which returns the performance to the design strategy for a new architecture. The whole process will end until such an optimal architecture that meets the need of the given task is found.

Understanding Task Properties. Each vision task has its intrinsic properties that would greatly determine what might be the optimal architecture of the deep network to build. Before going down to detailed network architecture design, one should make a thorough consideration of the given task properties, which includes but not limited to datasets size, data modalities, task objectives, and so on. For example, one can easily build a neural network for a standard imagery task by borrowing deep architectures that perform well on ImageNet benchmark. However, such a naive design might not be optimal if the size of datasets is quite smaller than ImageNet, which could potentially lead to over-fitting. These non-fitting defects could be even worse if researchers directly apply an ImageNet-scale deep architecture to new domains or modalities (*e.g.* fine-grained datasets, depth images, point clouds, and so on). Moreover, for some tasks with specific requirements of model efficiency in real-time deployment (*e.g.* high inference speed, low memory cost), some innovations of specific neural architectures are in desperate needed.

Manual or Automatic Design Strategy. Only by fully understanding given task properties, a proper design strategy could then be proposed. For a manual design strategy, it is greatly based on human expertise and experience. There are no clear principles towards guiding what exact strategy to choose or perform. Specifically, it starts with proposing hypotheses or ideas about addressing some particular needs in given tasks. These ideas or hypotheses are further concreted in newly designed candidate architectures.

One the other hand, the initial step of automatic design strategy is proposing a search space that allows candidate architectures can be selected properly. Such a search space with consideration of given task properties is often exponentially large or even unbounded. Then, a search strategy is introduced to explore the space of neural architectures, including random search, Bayesian optimisation, evolutionary methods, reinforcement learning (RL), and gradient-based methods.

Performance Evaluation. To find architectures that achieve high predictive performance on unseen data, the process of evaluating the performance for candidate architectures is essential. This process for manual designed architectures is quite conventional, which performs a standard training and validation of architecture on given datasets. The evaluated performance is returned to the design strategy for the next round of manual design with some modifications. However, for automatic design strategies, it is unfortunately computationally expensive and limits the number of architectures that can be explored. The solution is to estimate the performance with much lower computational demands. Thus, some practical methods would be needed, such as weight sharing, lower fidelity estimation, learning curve extrapolation, and network morphisms.

1.1.2 Challenges in Scalable Deep Learning Architecture Design

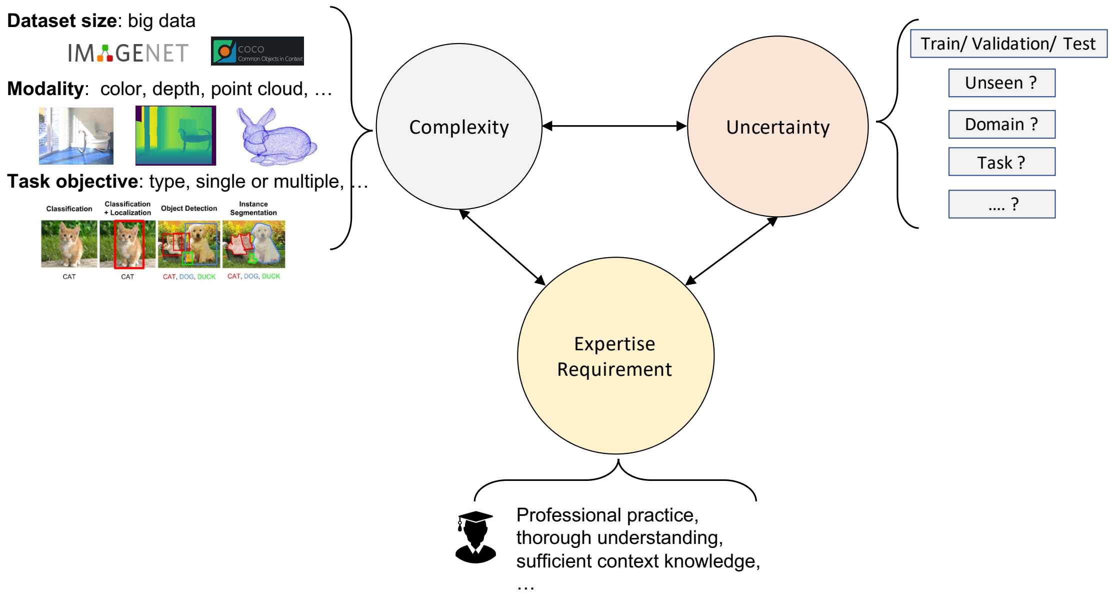


Figure 1.4: An abstract illustration of challenges in deep neural architecture design.

Despite such rapid progress in both manual and automatic paradigms, deep artificial neural architecture design for computer vision tasks remains challenging, when it comes considering scalability (the model size, learning capacity and efficiency, and domain adaptation to new data). Specifically, the challenges in scalable deep neural architecture design are summarised as follows:

Complexity. As described in Section 1.1.1, the initial procedure in designing deep learning architectures is understanding given task properties, which are inherent that greatly determines the potential form of the optimal architecture. However, these task properties—datasets size, data modalities, task objectives, and so on—are different from task to task. It is almost impossible to cover every aspect of a given task. Due to the complexity of these task properties, researchers have to go over through the design iterations, again and again, to gradually improve their understanding. This, in the end, results in a much tedious and laborious process of designing scalable neural network architectures.

Uncertainty. Generally, there are two types of uncertainty that bring difficulties to scalable deep learning architecture design: data uncertainty and task uncertainty. Due to data uncertainty, one fundamental objective of scalable deep learning architecture design is to find architectures that achieve high predictive performance on unseen data. However, we usually evaluate on a fixed set of training/validation data. It is intrinsically hard to prevent designed architectures would be overfitting. In term of task uncertainty, it is hard to guarantee that a network architecture specifically designed for a given task (*e.g*. ImageNet recognition) would be well-performed on a similar task (*e.g*. face recognition), not to mention a different task such as scene segmentation.

Expertise Requirement. The human expertise required by scalable deep learning architecture design is substantially high. It usually requires professional practices for a researcher to design a new deep learning architecture that meets the scalability requirements. More specifically, to propose a proper design strategy, a designer needs to be capable of thoroughly understand the inherent task properties for a given task. Furthermore, the designer should have sufficient context knowledge, *i.e*. being familiar with existing state-of-the-art networks in various challenging computer vision tasks, *e.g*. image classification, object detection and semantic segmentation, so as to bring some useful inspirations or atomic network units for a current design.

Overall, as illustrated in Figure 1.4, these three challenges in scalable deep learning architecture design are dependent. For instance, the complexity of dataset modality would scale the factor of data uncertainty, and vice versa, which would further require more expertise knowledge to tackle these challenges.

1.2 Contributions

The research of this thesis attempts to move steps further towards deep learning architecture design with consideration of scalability, studying approaches to designing deep neural architectures for several tasks by considering the underlying task characteristics for more efficient and powerful deep models. The contributions of this thesis to scalable deep learning architecture design research are summarised below:

1. **Chapter 3**: The under-studied model cost-effectiveness and scalability issue in deep learning person re-id is investigated, including model accuracy, inference cost, and matching efficiency. This differs substantially from the existing methods usually ignoring the model efficiency problem whilst only focusing on improving re-id accuracy rates. Through studying this problem, we aim for addressing large scale person re-id deployments typical in practical applications. A novel idea of jointly learning multi-granularity attention selection and feature representation is formulated for optimising person re-id cost-effectiveness in deep learning. This is the first attempt at jointly deep learning multiple complementary attention for solving the person re-id scalability problem. The proposed approach is

technically orthogonal to existing designs of efficient neural networks therefore allowing for implementing complementary strengths by concurrent integration in a hybrid architecture. Furthermore, a *Harmonious Attention Network* (HAN) framework is proposed to simultaneously learn hard region-level and soft pixel-level along with re-id feature representations for maximising the correlated complementary information between attention selection and feature discrimination in a compact architecture. This is achieved by devising an efficient Harmonious Attention module capable of efficiently and effectively learning different types of attention from the re-id feature representation hierarchy in a multi-task and end-to-end learning fashion.

2. **Chapter 4**: We investigate for the first time the scalability problem involved in person search. This is a fundamentally significant problem to be solved for scaling up the deep learning solutions to person search in the real-world applications. A *Hierarchical Distillation Learning* (HDL) approach is formulated for more discriminating knowledge transfer from a stronger teacher model into an efficient student model. A simple and effective teacher model is designed for joint learning of person search, which largely facilitates the knowledge distillation by avoiding knowledge transfer between structure inconsistent teacher and student models. Extensive experiments show the model cost-effectiveness and performance advantages of the HDL over the state-of-the-art alternative approaches on three person search benchmarks.

3. **Chapter 5**: A novel notion of Neural Graph Embedding (NGE) is proposed for NAS, characterised by jointly modelling the graphical topology of a network architecture and performing the network search in a continuous representation space. Introducing a neural graph concept and making a principled exploitation of Graph Convolutional Network, NGE addresses the limitation of the state-of-the-art methods in mining network topology knowledge, providing a generic neural architecture representation solution specially tailored for NAS. The proposed NGE method not only achieves highly competitive accuracy performance on CIFAR-10, CIFAR-100 and ImageNet, but also significantly reduces the architecture search process (taking only 0.1 GPU day for cell search). Moreover, the neural architecture discovered on CIFAR-10 by NGE can be readily *transferred* to the more challenging semantic segmentation task. The test with DeepLab-v3 on PASCAL VOC 2012 is performed and achieved 75.96% mIOU *without* the stronger COCO pretraining,

consistently outperforming the state-of-the-art network architectures.

4. **Chapter 6**: A novel heterogeneous search space for NAS is presented characterised by richer primitive operations including both conventional feature transformations and newly introduced feature self-calibration. This breaks the conventional selection limit of candidate neural networks and enables the NAS process to find stronger architectures, many of which are impossible to be discovered in the conventional space. This opens new territories for supporting stronger NAS algorithms and new possibilities for most expressive architectures ever to be revealed. A novel Neural Operator Search (NOS) method dedicated for NAS is formulated in the proposed heterogeneous search space, with a couple of key designs – heterogeneous residual block for fusing different types of tensor operations synergistically and attention guided search for facilitating the search process over a vast search space more efficiently and more effectively. With extensive comparisons to the state-of-the-art NAS methods, the experiments show that our approach is highly competitive on both CIFAR and ImageNet-mobile image classification tests.

1.3 Thesis Outline

The remaining chapters of this thesis are organised as follows:

Chapter 2 provides a review of existing research relevant to the main components of this thesis.

Chapter 3 proposes a harmonious attention network design for scalable person re- identification (re-id), enabling more discriminative re-id matching by efficient networks with more scalable inference in large scale re-id deployment scenarios with the need of processing a large amount of surveillance video data.

Chapter 4 proposes a hierarchical distillation network design for scalable per-son search, investigating the scalability problem of person search involving both model accuracy and inference efficiency simultaneously. It is fundamentally significant for real-world applications of person detection and person re-identification matching in unconstrained scene images.

Chapter 5 presents a neural graph embedding method for scalable neural architecture search, which leverages a Graph Convolutional Network to propagate and model the intrinsic topology information of network architectures. It is a generic neural network representation integrable with different existing NAS frameworks.

Chapter 6 presents a scalable neural operator search paradigm which unlocks existing limited feature-transformation-only search spaces, considering other advanced feature operations such as feature self-calibration by attention and dynamic convolutions. This new paradigm can search novel cell architectures with highly competitive performance.

Chapter 7 includes concluding remarks and discusses potential areas for future research and extensions.

Chapter 2

Literature Review

2.1 Handcrafted or Manually-Designed Deep Architectures

Substantial efforts have been made towards improving the performance on ImageNet benchmark by manually designing deep learning architectures. For example, Alex *et al.* propose a prototype of deep convolutional architecture (AlexNet) contains 5 convolutional layers, which outperforms the traditional classifiers trained on Fisher Vectors (FVs) computed from densely-sampled SIFT features [16]. ZFNet [17] improves the AlexNet by giving insight into the function of intermediate feature layers and the operation of the classifier using a novel visualization technique. VGGNet [4], on the other hand, investigates the effect of the convolutional network depth using an architecture with very small (3×3) convolution filters, demonstrating that a significant improvement can be achieved by pushing the depth to 16–19 weight layers. Based on the Hebbian principle and the intuition of multi-scale processing, GoogleNet [5] is composed of carefully crafted design inception modules, which allows for increasing the depth and width of the network while saving the computational budget. To resolve the degradation problem of deep models, ResNet [3] explicitly reformulates the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. This residual learning framework permits very deep networks with a depth of up to 152 layers—8× deeper than VGGNets but still having lower complexity. As shown in Figure 2.1, the evolution from AlexNet to ResNet involves introducing more layers and complicated structures. In general, these efforts were made with great demands for human knowledge and interventions.

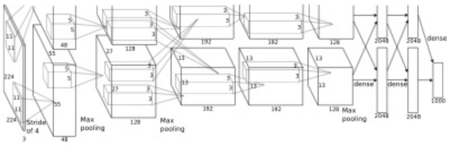
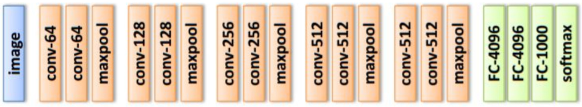
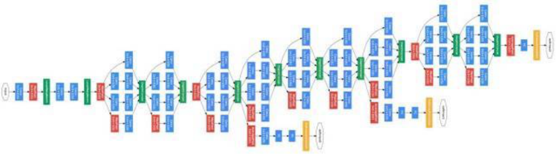
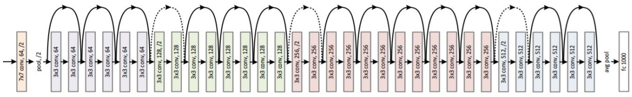
Handcrafted or Manually-Designed Deep Architectures	Features
 AlexNet	▪ 8 layers ▪ first breakthrough
 VGGNet	▪ 16,19 layers ▪ 3x3 small kernels
 GoogleNet	▪ 22 layers ▪ network in network
 ResNet	▪ 50, 101, 121 layers ▪ residual connections

Figure 2.1: Some examples of handcrafted or manually-designed deep architectures.

2.2 Neural Architecture Search (NAS)

Neural Architecture Search (NAS) is able to automate the tedious process of designing neural network architectures optimal for target tasks, bypassing the demand for rich domain knowledge and experiences. Recent attempts in NAS have achieved enormous success in various challenging tasks, e.g. image classification [6], object detection [7], and semantic segmentation [8, 9].

Categories in NAS. Since the seminal work by [6], neural architecture search has gained a surge of interest, effectively replacing laborious human designs by the computational process. From the ***strategy*** point of view, NAS methods can be categorised into two types: (1) proxy-based [6, 10, 11, 14] and (2) proxy-less [18, 19, 20] NAS. Specifically, to alleviate the computational cost during search, the proxy-based NAS methods search for building cells on proxy tasks, with one or more of following compromised strategies: starting with fewer cells; using a smaller dataset (e.g. CIFAR-10); learning with fewer epochs. Then, to transfer to the large-scale target task, one can build a network by stacking searched cells without further exploration. However, searched cells by proxy-based NAS methods are not guaranteed to be optimal on the target task. This is due to these search processes are normally not conducted on whole target datasets directly. In contrast, proxy-less NAS methods directly learns architectures on a target task by starting with an over-parameterised network (*supernet*) that contains all possible network connections, in which the redundant network connections are pruned to derive the optimised architecture. Notwithstanding significant better results than proxy-based approaches, proxy-less NAS methods require massive computational cost and GPU memory assumption, due to learning with the vast-size *supernet*. From the ***optimisation*** point of view, existing NAS methods usually fall into three groups: reinforcement learning (RL) based methods, evolutionary algorithm (EA) based methods, and gradient differentiable (GD) methods. In particular, RL-based NAS methods [6, 11, 19] control the selection of architecture component in a sequential order with policy networks. EA-based NAS methods [13, 12] employ the validation accuracies to guide the evolution of a population of initialised architectures. RL- an EA-based NAS methods usually suffer from low efficiency and high computational resource demand, due to the fundamental searching challenge in a discrete space. For instance, to search a state-of-the-art architecture for CIFAR-10 and ImageNet, it takes 2000 GPU days for RL [6] and 3150 GPU days for EA [13]. Several recent attempts have been made to improve, e.g. structural search space designing [12, 15], architecture weights sharing and inheritance [11, 21, 22]. Due to the fundamental searching challenge in a discrete space, RL and EA remain inefficient for NAS. In contrast, GD-based NAS methods [14, 23, 24] conduct searching over a continuous space by relaxation or mapping, substantially reducing the search cost to a few GPU days. For example, DARTS [14] simply relaxes the search space to be continuous by introducing a mixture of weights for all the candidate operations. NAO [24] maps a neural architecture into a continuous representation via an encoder model. Notwithstanding significantly lower search cost by further using weight sharing [11], both DARTS and NAO run the risk of being easily stuck around inferior local minimums as observed in [25]. SNAS [23] adopts the same continuity relaxation scheme as DARTS and probably shares the same limitation. Whilst varying in the algorithmic aspects, all these works commonly explore the *feature-transformation-only* search spaces without more diverse and advanced operations as investigated here. To show the NAS potential of the proposed richer search space with self-calibration learning operations, the efficient proxy-based GD optimisation is taken due to the resource constraint.

Operations in NAS. One common scheme for the standard proxy-based neural architecture search methods [11, 10, 14] is to factorise the search space via repeatedly stacking the same cell structure, within which a computing block generates an output tensor F_k by combining the transformations of two input feature tensors F_i and F_j as follows:

$$\mathsf{F}_k = o^{(i,k)}(\mathsf{F}_i) \oplus o^{(j,k)}(\mathsf{F}_j) \quad \text{s.t.} \quad i < k \& j < k, \tag{2.1}$$

where $o^{(i,k)}$ and $o^{(j,k)}$ are the i-th and j-th primitive operations for feature transformation, selected from a candidate operation set $\mathcal{O}$, and $\oplus$ is the element-wise addition. Existing NAS methods use only the standard *feature learning/transformation* operations (convolution, pooling and identity mapping) as the building components.

Besides, extensive studies [26, 27, 28, 29, 30, 31] have proven that other advanced operations for *feature self-calibration*, such as *attention learning* and *dynamic convolutions*, can bring great benefits for representation learning. For example, Hu *et al.* [26] proposes Squeeze-and-Excitation Networks to explicitly model inter-dependencies between channels by learning channel-wise self-attention. Jia *et al.* [29] presents Dynamic Filter Networks to generate context-aware filters for increasing the flexibility and adaptiveness of networks. However, these useful feature calibration elements have *never* been well exploited in NAS, significantly limiting the potentials of NAS which aims for automatically discovering more sophisticated and advanced network architectures without human engineering.

Architecture Space in NAS. Instead of an entire network architecture, a more feasible strategy is to search a repeatable structure [10], which factorises the search space via cells and blocks.

A cell consists of a set of N ordered *feature (tensor) nodes* $\{\mathsf{F}_k|, 1<=k<=N\}$. F_1 & F_2 are two *input nodes*, i.e. the outputs from the previous two cells. $\{\mathsf{F}_k\}_{k=3}^{N-1}$ denotes the *inner nodes* that perform computation. The *cell output* is the N-th node F_N, formed as the concatenation of all the inner nodes, i.e. $\mathsf{F}_N = \mathtt{concat}(\{\mathsf{F}_k\}_{k=3}^{N-1})$. There are two types of cells: *normal* cell (with stride of 1) and *reduction* cell (with stride of 2).

A block is defined as a *computational node* that outputs a feature node F_k (Fig. 2.2(a)) by transforming two input feature nodes F_i and F_j as:

$$\mathsf{F}_k = o^{(i,k)}(\mathsf{F}_i) + o^{(j,k)}(\mathsf{F}_j) \quad \text{s.t.} \quad i < k \;\&\; j < k, \tag{2.2}$$

where $o^{(i,k)}$ and $o^{(j,k)}$ are the i-th and j-th operations. Following DARTS [14], each operation is taken from the candidate set $\mathbb{O}$ with $O = 7$ primitive operations: (1) identity, (2) 3 × 3 max pooling, (3) 3 × 3 average pooling, (4) 3 × 3 separable convolution, (5) 5 × 5 separable convolution, (6) 3 × 3 dilated separable convolution, (7) 5 × 5 dilated separable convolution.

Generally, the architecture search space $\mathbb{A}$ is determined by the compositions of blocks, since the structure design of the input and output nodes in a cell is fixed. For a cell with $N = 7$ nodes, we only need to specify the inputs and operations for 4 *inner* computational nodes (blocks), resulting in a total number of $\prod_{n=1}^{N-3} \frac{(n+1)n}{2} \times O^2 \approx 10^9$ possible design choices.

Based on the definitions of cell and block above, one can construct a network in two steps: (i) Design a cell structure that contains $(N-3)$ ordered blocks; (ii) Stack multiple cells together. As shown in Fig. 2.2 (b), after the cell search is finished, M normal cells are stacked repeatedly for 3 times, interpolated with 2 reduction cells for downsampling the feature maps.

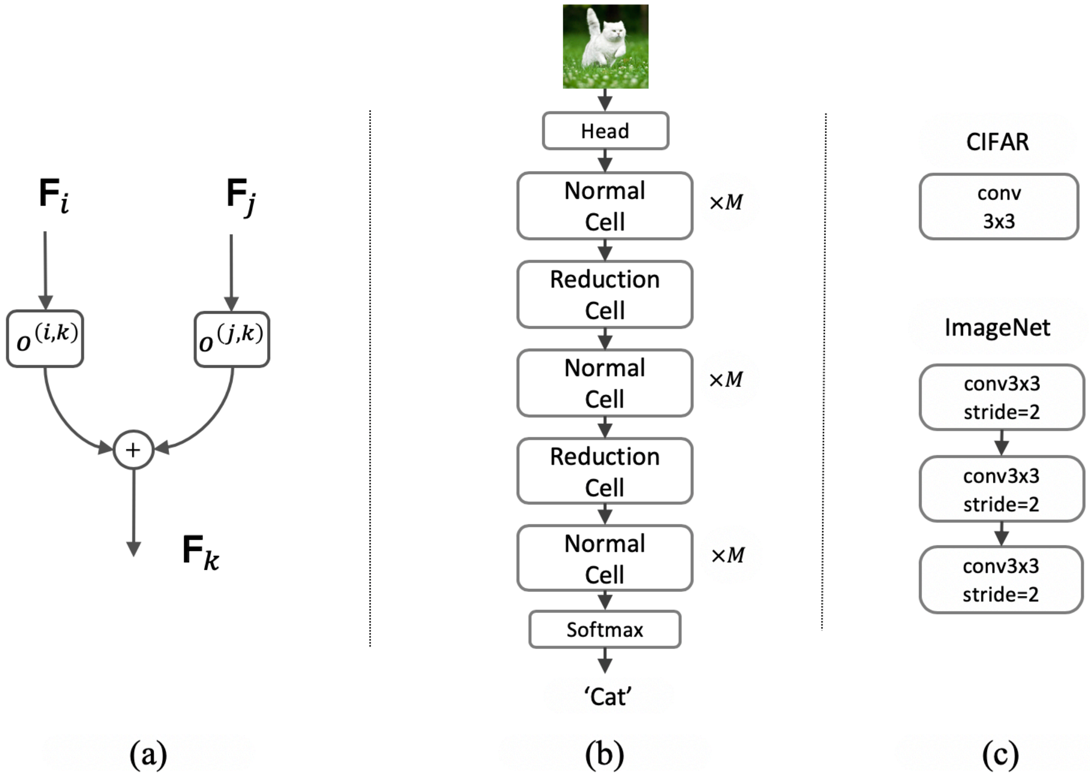


Figure 2.2: (a) The structure of the k-th block: taking two input feature tensors $\{F_i, F_j\}$, applying two separate operations $\{o^{(i,k)}, o^{(j,k)}\}$, and then combining them via element-wise addition as the output F_k. (b) Overview of building the network by stacking $M \times 3$ normal cells and 2 reduction cells. (c) The head architectures used for CIFAR and ImageNet.

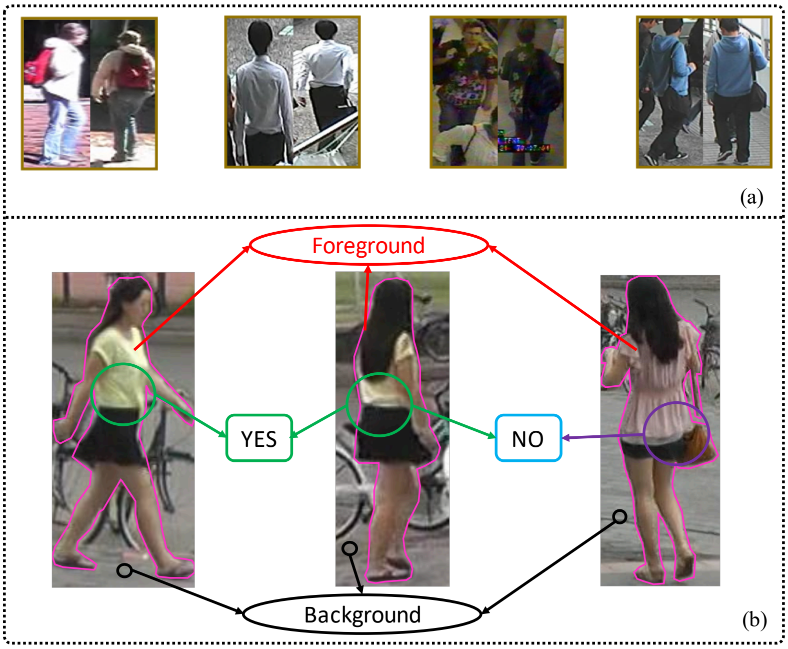


Figure 2.3: Person re-identification (re-id) is about **(a)** matching people across non-overlapping surveillance camera views which requires **(b)** capturing discriminative (attentional) parts and distinguishing foreground and background regions in person images.

2.3 The Person Re-Identification Problem

Person re-identification (re-id) aims to search people across non-overlapping surveillance camera views deployed at different locations by matching auto-detected person bounding box images (Figure 2.3).

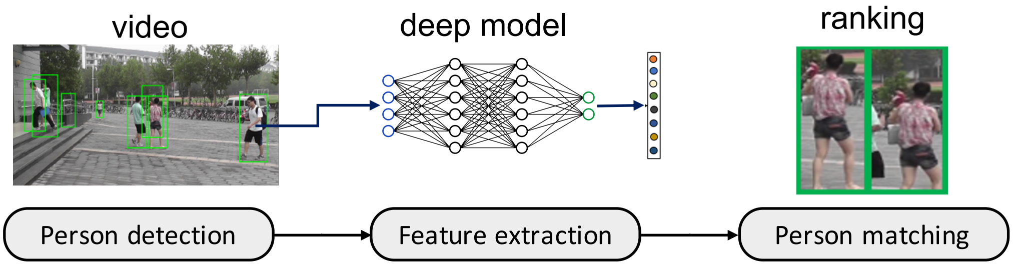


Figure 2.4: The pipeline of a person re-identification system.

Person Re-id System. As shown in Figure 2.4, a standard pipeline of a person re-identification system consists of three steps: person detection, person feature extraction, and person matching. Specifically, in the first step, a large pool of person images are generated by applying state-of-the-art person detectors [32, 33] on the raw video frames collected by surveillance cameras. Then, to extract discriminative visual features to describe individual appearances for each person image, the typical way is using deep models [4, 5, 3]. After feature extraction, matching the imagery features of the query (or interchangeably termed as probe) images against a gallery of persons is performed by measuring the similarity between features, such as euclidien, l_1 and cosine distance. Given that the research community treats the person detection as independent research areas, in this thesis, we focus on studying the deep re-id feature extraction.

Scalability. With the 24/7 operating nature of surveillance cameras, person re-id is *intrinsically* a large scale search problem with a fundamental requirement for developing systems with both *fast data throughput* (i.e. low inference cost) and *high matching accuracy*. This is because, model accuracy and inference efficiency *both* are key enabling factors for affordable real-world person re-id applications. In this thesis, we define this *cost-effectiveness* measure as the *scalability* of a person re-id system, taking into account model accuracy and computational cost *jointly*, *rather than optimising either alone*. Earlier person re-id methods in the literature rely on slow-to-compute high-dimensional hand crafted features with inferior model performance, yielding unsatisfactory solutions [34, 35, 36, 37, 38].

The recent introduction of large scale person re-id datasets [39, 40, 41, 42] allows for a natural utilisation of increasingly powerful deep neural networks [3, 43, 44], substantially improving person re-id accuracy in a single system pipeline. However, typical existing deep learning re-id methods remain *large sized* and *computationally expensive* therefore unfavourable for real deployments in scalability. This is due to the adoption of deep and wide neural network architectures with a huge number of parameters and exhaustive multiply-add operations. For example, the often-selected CNN architecture ResNet50 [3] consists of 25.1 million parameters consuming 3.80×10^9 FLoating-point OPerations (FLOPs) in forwarding a single person image through the network. While the offline training of large neural networks can be reasonably afforded using industrial-sized or cloud computing clusters with rich high-performance graphics processing units (GPUs), deploying them to process *big video data* suffers from low inference efficiency and expensive energy consumption. There is an intrinsic need for designing cost-effective deep learning re-id methods, which is currently less investigated with insufficient research efforts and attempts.

Re-id Feature Learning. The state-of-the-art person re-id deep methods are typically concerned with supervised learning of identity-discriminative representations [45, 46, 47, 48, 49, 50, 51, 52, 53, 39]. Although unsupervised learning and transfer learning based techniques are progressively advancing [54, 55, 56, 57, 58, 59, 60], their re-id performances are significantly inferior therefore less satisfactory and reliable in practical use. With the emergence of large benchmark datasets [41, 40, 61, 39], more powerful and computationally expensive neural networks like ResNet50 [3], originally designed for object image classification, have been increasingly adopted in building person re-id model architectures. The use of stronger and heavier networks yields significant gains in performance, but simultaneously sacrifices largely the deployment efficiency due to the need for high memory and computing consumption apart from lengthy model inference. Such inefficient systems suffer from low data throughput, therefore limiting the possible application scenarios (*undesired* in processing a large pool of surveillance videos). One intuitive approach to large scale person re-id is to train efficient small neural network models. This is made more attractive by the development of lightweight architectures, e.g. MobileNet [62], ShuffleNet [63], and CondenseNet [64]. These methods are based on the observation that there exist highly redundant weights in large neural networks [65]. However, such networks, originally designed for generic object classification and detection, are less effective for visually fine-grained and subtle person re-id matching. It is non-trivial to *simultaneously* achieve both generalisation performance and inference efficiency by a single deep learning person re-id model.

Model Efficiency. In the literature, model efficiency is an under-studied and critical problem in person re-id. Zheng *et al.* [40] employed a KD-tree based approximate nearest neighbour (ANN) method to expedite the re-id matching process. As another ANN strategy, the learning-to-hash idea has been explored with hand-crafted [66] and deep learning [67, 68] models. These methods quantise the feature representations so that the hamming distance metric can be applied to rapidly compute matching scores at the cost of significant performance degradation due to limited expressive capacity. Recently, Wang *et al.* [69] proposed to conduct re-id matching subject to given computation budgets. The hypothesis is that feature representations of easy samples can be computed at lower costs which makes room for computation reduction. However, it is intrinsically difficult and ambiguous to measure the sample easiness degree given the poor-quality surveillance data and the nature of pairwise matching (*not* per-sample inference).

Unlike all these existing strategies, we explore differently the potential of person attention learning for model efficiency and cost-effectiveness. Conceptually, our method is complementary to the prior techniques with extra possible performance benefits.

Attention Learning. There exist *learning-to-attend* algorithms developed for improving re-id particularly in misaligned person bounding boxes, e.g. those generated by automatic detection. Earlier approaches are based on localised patch matching [70, 71] and saliency weighting [72, 73]. These solutions are mostly ineffective to cope with poorly aligned person images, due to the stringent requirement of tight bounding boxes around the whole person and high dependence on weak hand-crafted features. Besides, such algorithms are usually more computationally expensive with a need for explicit and complex patch processing.

To overcome the aforementioned limitation, more advanced re-id attention deep learning methods have been recently proposed [74, 75, 76, 77, 78, 79, 80, 81]. A common strategy taken by these methods is to incorporate a regional attention (i.e. *hard attention*) selection sub-network into a deep re-id model. For example, Su *et al.* [76] integrated a pose detection model separately learned from auxiliary pose ground-truth into a part-based re-id model. Li *et al.* [74] designed an end-to-end trainable part-aligning CNN for extracting latent discriminative regions and exploiting these regional features to perform re-id. Zhao *et al.* [75] exploited a Spatial Transformer Network [82] as a hard attention module to search re-id discriminative parts given a pre-defined spatial constraint. Lan *et al.*[77] formulated a reinforcement attention selection model for salient region refinement under identity discriminative constraints. Qian *et al.* [80] rotated persons to canonical poses through pose-specific image synthesising.

A common weakness of these above models is the lack of handling noisy pixel information within selected regions, i.e. no *soft attention* modelling. This issue was considered in [83]. However, this model assumes tight person bounding boxes therefore not suitable for processing poor detections. In parallel to this thesis, Xu *et al.* [78] considered a joint end-to-end learning of both body parts and regional saliency. Along with continuously improved matching performance, all the attention learning re-id methods come with significantly increased model complexity and inference costs for realising strong model generalisation capability. This dramatically limits their scalability and usability in large scale re-id deployments.

2.4 The Person Search Problem

Person search considers the problems of person detection and person re-identification (re-id) *simultaneously* [84, 85]. It is valid and necessary due to that the practical application of person re-id relies heavily on person detection. The detection quality of persons on the surveillance scene images affects the re-id performance largely. For example, missing detection causes the *inability* of person re-id on the corresponding person instance, and misalignment introduces *noise*

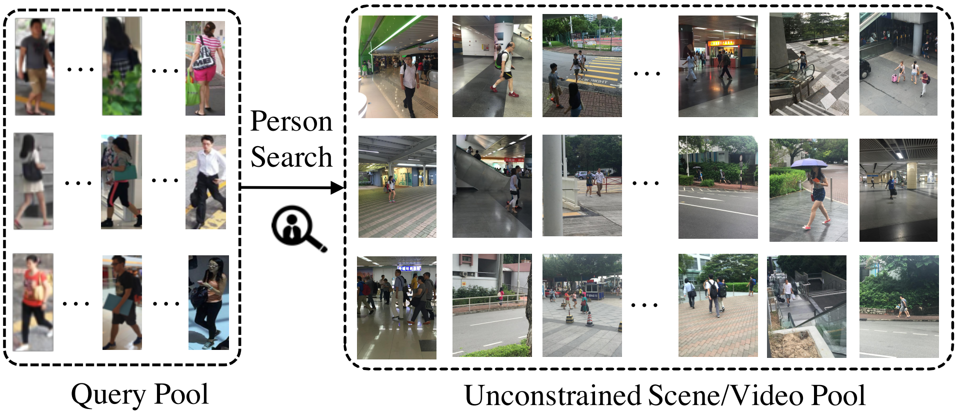


Figure 2.5: The significance of scalability in person search. Both sets of query persons and scene imagery are of large scale.

or *information loss* to person re-id.

In addition to person matching accuracy, this task joining by person search also expands the scope for model efficiency considerations. Conventionally, person search efficiency is mostly considered in person re-id model design, since person bounding boxes are assumed already available. This breaks the connection between person re-id and person detection, therefore, losing their joint computing opportunity for improving model efficiency. This issue is naturally solved in the person search problem setting.

Model efficiency is fundamentally crucial for *scalable* person search, due to the intrinsic large scale search requirement in real-world deployment (Figure 2.5). The efficiency problem was initially investigated in the introduction of person search [84], followed by a few followup *joint learning* model designs [86, 87, 88]. However, all these existing methods are significantly outperformed by *independent learning* competitors [89, 90].

Moreover, some of the joint learning methods [86, 88, 91] are even *not necessarily* more efficient than independent learning, because of their query-specific search design nature. That is, the model needs to conduct an independent search process in *every* whole scene image for *every* query person, with the search cost proportional to the *quadratic pairwise combination* (i.e. multiplication) of the query and gallery samples. This implies potentially even *more inefficient* solutions than simpler independent learning [86, 88], totally opposite to their original efficiency objective.

In the literature, only the OIM method [84] makes an initial attempt for efficient person search. The key idea is that person detection and person re-id can share a large proportion of computing cost by jointly using the low-level feature network layers. This is analogous to the core idea of Faster R-CNN [32]. After the OIM model is trained, person detection and re-id feature extraction can be conducted jointly on the gallery data by a single network. It is a *one-off* process, independent to the size of query images therefore much more scalable than query-specific search models. However, the main focus of OIM is on how to exploit unlabelled person instances for improving re-id matching. This method does not fully investigate the significant model efficiency problem. This is partly due to that its performance is somewhat weak, e.g. significantly inferior to the current state-of-the-art methods [89, 90]. Overall, the *scalability* problem including both *model accuracy* and *inference efficiency* for person search remains largely under-studied, despite its significant practical importance.

Due to a more comprehensive problem formulation, person search has gained increasingly more attention and research efforts [89, 90, 86, 88, 87] since its establishment [84, 85]. Existing methods are generally fallen into two groups: (1) *independent learning* (IL) [89, 90] and (2) *joint learning* (JL) [84, 86, 88, 87, 91, 92] based models.

Thus far, the *independent learning* based person search methods achieve the state-of-the-art performance [85, 89, 90]. They separate person detection and re-id matching by designing independent network models. Strong and computationally expensive CNN models [3] are often selected in such designs for maximising the search accuracy. One of the major disadvantages for these methods is costly deployment and slow execution. The model inference efficiency can be further reduced due to the addition of auxiliary components such as foreground segmentation and multi-branch fusion [90]. Although reaching good performance, this group of methods are less scalable computationally therefore unsuitable for large scale deployments typically required in real applications.

The *joint learning* based person search methods have been developed with one of the main objectives as solving the above efficiency limitation [84, 86, 88, 87, 91, 92]. The methods in [84, 87] improve the model inference speed by taking advantages of the Faster R-CNN design. The key idea is to make person detection and re-id tasks share the low-level feature computation. NPSM [88], RCAA [86] and QEEPS [91] suggest query-specific person search strategies. CGPS [92] learns contextual graph representations via coupling the targets and the background contexts. Opposite to their design objectives for efficiency gain, these existing models *all* suffer from another scalability limitation: every query-gallery pair needs to be processed independently.

This means that the detection cost is proportional to the combination of query and gallery samples. Instead, person detection on all gallery images is conducted *one-off* in [84, 87], therefore independent and scalable to any sizes of query tasks. The efficiency of NPSM [88] is also significantly limited by the need of generating region proposals, e.g. EdgeBox [93]. Besides, all these models are often less powerful than the counterparts.

2.5 Knowledge Distillation

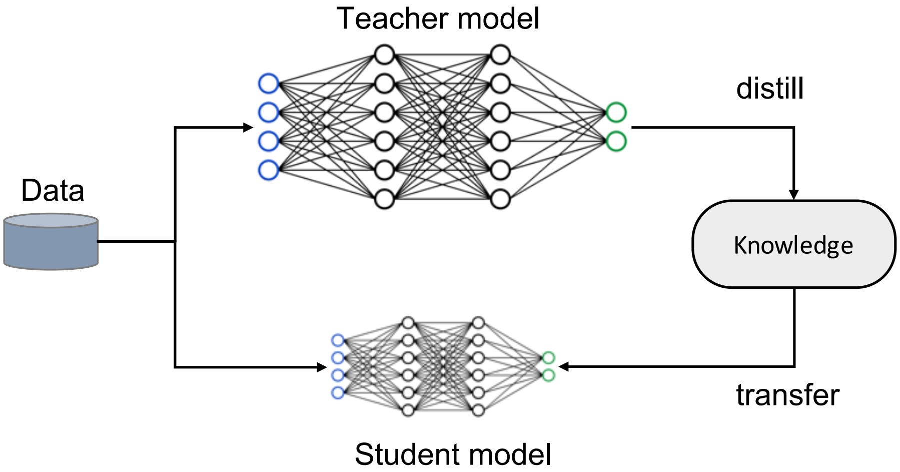


Figure 2.6: A generic teacher-student framework for knowledge distillation.

There are recent works that apply knowledge distillation for computer vision and natural language processing (NLP) problems. The core idea is that small student models learn the knowledge in big teacher models to achieve a competitive or even superior performance. A general teacher-student framework for knowledge distillation is shown in Figure 2.6. Normally, different types of knowledge are usually considered for distillation. Specifically, the predictions (logits) of a large deep model [94] are vanilla knowledge to guide the learning of the student model. Also, features (*i.e.* activations or neurons) of intermediate layers [95] contain the rich information learned by teacher model that can be used as the typical knowledge. Other forms of intermediates such as attentions [96] can further enrich the categories in knowledge distillation.

In contrast to all the existing person search methods, we consider the scalability and cost-effectiveness problem of person search including both model accuracy and inference efficiency. None of the previous methods are designed to address this problem, lacking sufficient model generalisation and/or inference efficiency. To this end, we develop a simple and strong joint learning model that reaches the performance of the state-of-the-art independent learning method.

This layouts a very competitive baseline method and inspires novel ideas to the future works.

In a NAS setting, we use knowledge distillation for closing the performance gap between a proxy network and a full network on target task for the proxy-based NAS. In particular, we aim to incorporate self-calibration learning in the context of NAS, which motivates us to leverage attention transfer [96] to mimic the hidden attention maps in a pre-trained model.

Chapter 3

Scalable Attention Network Design

3.1 Overview

In this chapter, we investigate *the under-studied scalability and cost-effectiveness problem in deep learning person re-identification*. To this end, we explore the potential of *person attention selection learning* in a single neural network architecture. The rationale is that detecting the fine-grained salient parts of person images not only allows to preserve the model matching performance, but also favourably simplifies the re-id matching due to noise suppression, therefore *rendering small networks sufficient* to induce this simplified target matching function. It is this re-id attention selection learning strategy that distinguishes our method from existing purpose-generic network compression techniques [62, 63, 64], enabling uniquely a simultaneous realisation of model efficiency and generalisation performance. Owing to the conceptual orthogonality, existing network compression techniques can be naturally integrated as complementary designs into our approach to achieve further model efficiency and scalability.

There have been a number of existing attempts at learning re-id attention selection. Nevertheless, their primary purpose is to address the person misalignment issue for higher model generalisation capability. This is because in practical re-id scenarios, person images are usually automatically detected with arbitrary cropping errors for scaling up to large video data [40, 41, 42]. Additionally, people are often captured in various poses across open space and time. There is consequently an inevitable need for attention selection within arbitrarily-aligned bounding boxes as an integral part of model learning for re-id.

A common earlier strategy for re-id attention selection is local patch calibration and saliency weighting in pairwise image matching [73, 70, 71, 72]. This approach relies on pre-fixed hand-crafted features without deep learning jointly more expressive representations and a matching metric in an end-to-end manner. A number of more advanced attention deep learning models for person re-id have been recently developed [74, 75, 76, 77, 78, 79, 80, 81]. Most of these methods consider only coarse region-level attention whilst ignoring the fine-grained pixel-level saliency. Moreover, such methods depend on heavy network architectures therefore suffering the drawbacks of high computational complexity and low model inference efficiency. This thesis addresses the weaknesses and limitations of these existing methods for scalable person re-id with both superior matching accuracy and inference efficiency.

Beyond the conventional advantage from the attentional weighing for more discriminative person matching, the proposed approach is specially designed to simultaneously address the efficiency weaknesses of existing re-id attention methods. This is achieved by formulating a novel attention module that enables a efficient joint learning of both soft and hard attention in compact CNN architectures *harmoniously* whilst preserving the model generalisation capability. The results of this design are a class of cost-effective harmonious attention networks dedicated for scalable re-id matching with state-of-the-art accuracy performance. This is the first attempt of modelling multi-level correlated attention in deep learning for person re-id to our knowledge. In addition, we introduce cross-attention interaction learning for further enhancing the complementary effect between different levels of attention subject to re-id discriminative constraints. This is impossible to do for most existing methods due to their inherent single level attention modelling design. We show the benefits of joint modelling multi-level attention in person re-id in our experiments.

3.2 Scalable Person Re-Identification

Problem Definition. Suppose there are n training bounding box images $\mathcal{I} = \{\boldsymbol{I}_i\}_{i=1}^{n}$ from n_{id} distinct people not being seen in more than one view simultaneously together with the corresponding identity class labels $\mathcal{Y} = \{y_i\}_{i=1}^{n}$ where $y_i \in [1, \cdots, n_{\text{id}}]$. We aim to learn a deep feature representation model optimal for person re-id matching under significant viewing condition variations with high computational efficiency. To that end, we formulate a *lightweight* (less parameters and multiplication-addition operations) ***Harmonious Attention Network*** (HAN). This

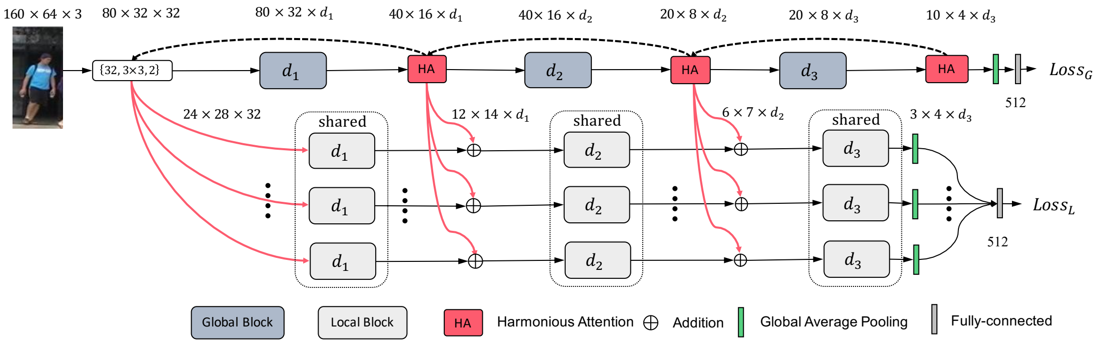


Figure 3.1: Schematic architecture of the proposed Harmonious Attention Network (HAN). The design of HAN is characterised by high cost-effectiveness for maximising the model scalability. This is enabled by the introduction of a lightweight but effective Harmonious Attention (HA) module (see Fig. 3.2 and Fig. 3.3) and a computationally efficient depthwise separable convolution based building block (see Fig. 3.4 and Table 3.1). The symbol d_l ($l \in \{1,2,3\}$) denotes the number of convolutional filter in the corresponding l-th block.

takes a principle of attention learning particularly for achieving cost-effective person re-id. The objective of HAN is to concurrently learn a set of harmonious attention along with both global and local feature representations for maximising their complementary benefit and compatibility between the model components in terms of both the discriminative capability and inference efficiency.

Formulation Rationale. The HAN model design is based on two motivating considerations: (1) The human visual system that leverages both global contextual and local saliency information concurrently in conjunction with the evolution attention search capability [97, 98]; (2) The divide-and-conquer algorithm design principle [99] that decomposes the re-id feature learning task at different levels of granularity (global & local) and significance (salient or not), simplifying the target problem formulation and enabling efficient small networks suffice to model the desired representations. Intuitively, joint learning of global-local feature representations with attention extracts correlated complementary information in different context, hence efficiently achieving more reliable recognition due to such selective discrimination learning. For bounding box image based re-id, we consider the entire person in the image as a *global scene context* and body parts of the person as *local information sources*, *both* subject to the surrounding background clutter, potentially also misalignment and partial occlusion due to poor detections. Typically, the location information of body parts is not provided in person re-id image annotation, i.e. only weakly labelled without fine-grained part labels. Therefore, the person attention learning is a *weakly supervised* learning task in the context of optimising the re-id performance. Under the global-local concurrent design, we consider a multi-branch network architecture. The overall objective of this multi-branch scheme and the architecture composition is to minimise the target function modelling complexity in a divide-and-conquer manner. This enables reducing the network parameter size whilst still maintaining the model representation learning capacity.

HAN Overview. The overall design of our HAN architecture is depicted in Fig. 3.1. In particular, the attention model contains two branches with hierarchically distributed attention modules: **(1)** One *local branch* consisting of T streams with an identical structure: Each stream aims to learn the most discriminative visual features for one of T local regions of a person bounding box image. To reduce the model parameter size, we share the layer parameters among all local streams. **(2)** One *global branch*: This aims to learn the optimal global level features from the entire person images. **(3)** Hierarchical *harmonious attention learning*: This aims to discover and exploit re-id discriminant saliency regions (hard attention) and pixels (soft attention) concurrently in a synergistic interaction with global and local feature representations in an end-to-end learning manner. Next, the main designs of the proposed HAN model is described.

3.2.1 Global-Local Feature Learning

The HAN model is designed to perform *global-local representation learning subject to the same identity label constraints* by allocating each branch with a separate objective loss function derived from the per-batch training person classes. As a consequence, the learning behaviour of each branch is conditioned respectively on their own feature representations.

Loss Function. For model training, we use the softmax cross-entropy loss function. Formally, we start by predicting the class posterior probability $\tilde{y}_i$ of a person image $\boldsymbol{I}_i$ over the ground-truth identity class label y_i:

$$p(\tilde{y}_i = y_i|\boldsymbol{I}_i) = \frac{\exp(\boldsymbol{w}_{y_i}^\top \boldsymbol{x}_i)}{\sum_{k=1}^{|n_\text{id}|} \exp(\boldsymbol{w}_k^\top \boldsymbol{x}_i)} \tag{3.1}$$

where $\boldsymbol{x}_i$ refers to the feature vector of $\boldsymbol{I}_i$ from the corresponding branch, and $\boldsymbol{w}_k$ the prediction function parameter of training identity class k. The cross-entropy loss for a mini-batch of n_bs training images is then defined as:

$$\mathcal{L}_\text{ce} = -\frac{1}{n_\text{bs}} \sum_{i=1}^{n_\text{bs}} \log \Big(p(\tilde{y}_i = y_i|\boldsymbol{I}_i) \Big) \tag{3.2}$$

Sharing Low-Level Feature Learning. In a HAN model, we construct the global and local branches on a common low-level conv layer, in particular the first conv layer. This is for facilitating the purpose of inter-branch common representation learning (Fig. 3.1). The intuition is that, the bottom conv layer captures elementary features such as edges and corners shared by both global and local patterns of person images. This design is in spirit to multi-task learning [100], where the local and global feature learning branches are viewed as two correlated learning tasks. Besides, sharing the low-level layer reduces the model parameter size, not only mitigating the model overfitting risk but also improving the model inference efficiency. This is critical in learning person re-id models especially when the labelled training data is limited.

Attention in Hierarchy. We take a *block-wise* attention module design, that is, each attention module is specifically optimised to attend the input feature representations at its own level *alone*. In the CNN hierarchical framework, this naturally allows for *hierarchical* multi-level attention learning to progressively refine the attention maps, again in a spirit of the divide-and-conquer design [99]. As a result, we can significantly reduce the attention search space (i.e. the model optimisation complexity) whilst allowing multi-scale selectiveness of hierarchical features to enrich the final feature representations.

Such progressive and holistic attention modelling is intuitive and essential for re-id due to that (1) the surveillance person images often have cluttered background and uncontrolled appearance variations therefore the optimal attention patterns of different images can be highly varying, and (2) a re-id model typically needs robust (generalisable) model learning given very limited training data (significantly less than common image classification tasks).

Unlike most existing attention selection based person re-id works that simply adopt a standard CNN network with a large number of model parameters and high computational cost in model deployment [1, 5, 3, 78], our HAN design is more efficient (faster inference in deployment) whilst still having deep CNN architectures to maintain strong discriminative power. This is particularly critical for re-id where the label data is often sparse (large models are more likely to overfit in training) and the deployment efficiency is important for practical applications at scales (slow feature extraction is not scalable to large surveillance video data).

Remarks. The HAN model aims to learn concurrently multiple re-id discriminative feature representations for different local image regions and the entire image. All these representations are optimised by maximising the *same* identity classification tasks individually and collectively at the same time. Concurrent multi-task learning in a multi-loss design enables to preserve both local saliency in feature selection and global coverage in image representation.

In terms of loss function design, while many existing person re-id methods [101, 52, 69, 102, 103, 104, 105, 106, 41] suggest the importance of using pairwise comparison based loss objectives, e.g. the triplet and contrastive functions, we empirically found that the simpler cross-entropy loss suffices to achieve satisfied discriminative learning without any extra complexity introduced from hard sample mining. We partly attribute this to the strong capability of the HAN model in automatically attending re-id discriminative information, simplifying the loss design complexity.

Importantly, using only the classification loss formulation brings about a couple of practical benefits: **(i)** This significantly *simplifies* the training mini-batch data construction, e.g. only random sampling without any tricks required. This makes our HAN model more scalable to situations when very large training population is available. **(ii)** This also eliminates the *undesirable* need for carefully forming pairs and/or triplets in preparing re-id training samples, as in these existing methods, due to the inherent imbalanced negative and positive pair size distributions.

We consider that the key to person re-id is about model generalisation to unseen test identity classes given the training data from *independent* seen classes. This loss choice is supported by previous visual psychophysical findings that representations optimised for classification tasks generalise well to novel categories [107]. We exploit this general classification learning principle beyond the stringent pairwise relative verification loss designs.

3.2.2 Harmonious Attention Learning

To perform attention selection within person bounding box images with unknown misalignment, we formulate a *harmonious attention learning* scheme. This is the core module component of the proposed model. Specifically, this scheme jointly learns a collection of complementary attention maps, including hard (regional) attention for the local branch and soft (spatial/pixel-level and channel/scale-level) attention for the global branch. Besides, we introduce a *cross-attention interaction learning* scheme between the local and global branches for further enhancing the harmony and compatibility degree whilst simultaneously optimising per-branch discriminative feature representations. Next, we describe the design details of the Harmonious Attention Module.

(I) Soft Spatial-Channel Attention. The input to a Harmonious Attention module is a 3-D tensor $\boldsymbol{X}^l \in \mathcal{R}^{h \times w \times c}$ where h, w, and c denote the number of pixel in the height, width, and channel

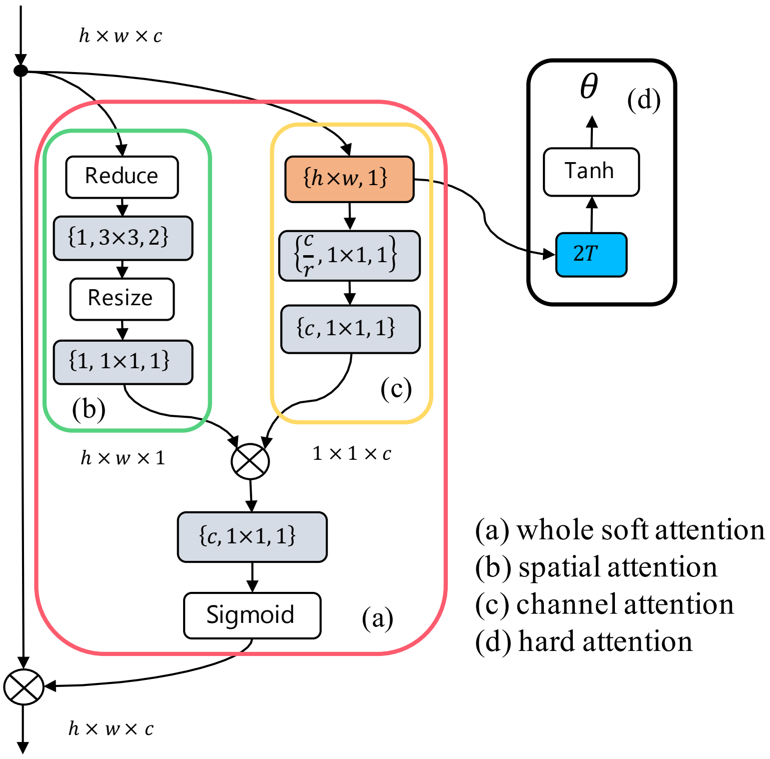


Figure 3.2: Structure of a Harmonious Attention module consists of **(a)** Soft Attention which includes **(b)** Spatial Attention (pixel-wise) and **(c)** Channel Attention (scale-wise), and **(d)** Hard Regional Attention (part-wise). Layer type is indicated by background colour: **grey** for *convolutional* (conv), **brown** for *global average pooling*, and **blue** for *fully-connected* layers. The three items in the bracket of a conv layer are: filter number, filter shape, and stride. ReLU [1] and Batch Normalisation [108] (applied to each conv layer) are not shown for brevity.

dimensions respectively; and l indicates the level of this module in the entire network (multiple such modules). Soft spatial-channel attention learning aims to produce a saliency weight map $\boldsymbol{A}^l \in \mathcal{R}^{h \times w \times c}$ of the same size as $\boldsymbol{X}$.

Given the largely independent nature between spatial (inter-pixel) and channel (inter-scale) attention, we propose to learn them in a *joint* but *factorised* way as:

$$\boldsymbol{A}^l = \boldsymbol{S}^l \times \boldsymbol{C}^l \tag{3.3}$$

where $\boldsymbol{S}^l \in \mathcal{R}^{h \times w \times 1}$ and $\boldsymbol{C}^l \in \mathcal{R}^{1 \times 1 \times c}$ represent the spatial and channel attention maps, respectively.

We perform the attention tensor factorisation by designing a two-branches unit (Fig. 3.2(a)): One branch to model the spatial attention $\boldsymbol{S}^l$ (shared across the channel dimension), and another branch to model the channel attention $\boldsymbol{C}^l$ (shared across both height and width dimensions). By this design, we can compute *efficiently* the full soft attention $\boldsymbol{A}^l$ from $\boldsymbol{C}^l$ and $\boldsymbol{S}^l$ with a tensor multiplication. Our design is more efficient than common tensor factorisation algorithms [109] since heavy matrix operations are eliminated.

(i) Spatial Attention. We model the spatial attention by a tiny (10 parameters) 4-layers sub-network (Fig. 3.2(b)). It consists of a global cross-channel averaging pooling layer (0 parameter), a conv layer of 3 × 3 filter with stride 2 (9 parameters), a resizing bilinear layer (0 parameter), and a scaling conv layer (1 parameter). In particular, the global average pooling, formulated as

$$S^l_{\text{input}} = \frac{1}{c}\sum_{i=1}^{c} \boldsymbol{X}^l_{1:h,1:w,i} \tag{3.4}$$

is designed especially to compress the input size of the subsequent conv layer with merely $\frac{1}{c}$ times of parameters needed. This cross-channel pooling is reasonable because in our design all channels share the identical spatial attention map. We finally add a scaling layer for automatically learning an adaptive fusion scale in order to optimally combining the channel attention described next.

(ii) Channel Attention. We model the channel attention by a small 4-layers squeeze-and-excitation component (Fig. 3.2(c)). We first perform a *squeeze* operation via an averaging pooling layer (0 parameters) to aggregate the feature information distributed across the spatial space into a channel signature as

$$C^l_{\text{input}} = \frac{1}{h \cdot w}\sum_{i=1}^{h}\sum_{j=1}^{w} \boldsymbol{X}^l_{i,j,1:c} \tag{3.5}$$

This signature conveys the per-channel filter response from the whole image, therefore providing the complete information for the inter-channel dependency modelling in the subsequent *excitation* operation, formulated as

$$C^l_{\text{excitation}} = \texttt{ReLU}\big(\boldsymbol{W}^{\text{ca}}_2 \times \texttt{ReLU}(\boldsymbol{W}^{\text{ca}}_1 C^l_{\text{input}})\big) \tag{3.6}$$

where $\boldsymbol{W}^{\text{ca}}_1 \in \mathcal{R}^{\frac{c}{r}\times c}$ ($\frac{c^2}{r}$ parameters) and $\boldsymbol{W}^{\text{ca}}_2 \in \mathcal{R}^{c\times\frac{c}{r}}$ ($\frac{c^2}{r}$ parameters) denote the parameter matrix of 2 conv layers in order respectively, and r (16 in our implementation) represents the bottleneck reduction rate. This bottleneck design reduces the model parameter number from c^2 (using 1 conv layer) to $(\frac{c^2}{r}+\frac{c^2}{r})$, e.g. only need $\frac{1}{8}$ parameters when r=16.

For facilitating the combination of the spatial attention and channel attention, we further deploy a $1\times1\times c$ conv (c^2 parameters) layer to compute blended full soft attention after tensor multiplication. This is because the spatial and channel attention are not mutually exclusive but with a co-occurring complementary relationship. Finally, we use a sigmoid operation (0 parameter) to normalise the full soft attention into the range between 0.5 and 1.

Remarks. Our model is similar to Residual Attention (RA) [28] and Squeeze-and-Excitation (SE) [110] concepts but with a number of essential differences: **(1)** The RA requires to learn a much more complex soft attention sub-network which is not only computationally expensive but also less discriminative when the training data size is small typical in person re-id. **(2)** The SE considers only the channel attention and implicitly assumes non-cluttered background, therefore significantly restricting its suitability to re-id tasks under cluttered surveillance viewing conditions. **(3)** Both RA and SE consider no hard regional attention modelling, hence lacking the ability to discover the correlated complementary benefit between soft and hard attention learning.

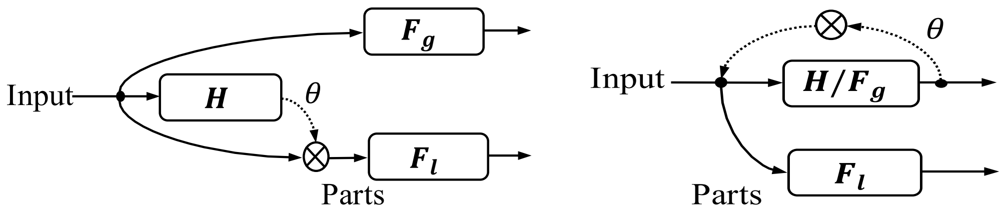
(II) Hard Regional Attention. The hard attention learning aims to locate latent (*weakly supervised*) discriminative T regions (e.g. human body parts) in each input image at the l-th level. We model this regional attention by learning a transformation matrix as:

$$\boldsymbol{A}^l = \begin{bmatrix} s_h & 0 & t_x \\ 0 & s_w & t_y \end{bmatrix} \tag{3.7}$$

which enables image cropping, translation, and isotropic scaling operations by varying two scale factors (s_h, s_w) and the 2-D spatial position (t_x, t_y). We pre-define the region size by fixing s_h and s_w for limiting the model complexity. Therefore, the effective modelling part of $\boldsymbol{A}^l$ is only t_x and t_y, with the output dimension as $2\times T$ (T the region number).

To perform this learning, we introduce a simple 2-layers ($2\times T\times c$ parameters) sub-network (Fig. 3.2(d)). We exploit the first layer output (a c-D vector) of the channel attention (Eq. equation 3.5) as the first FC layer ($2\times T\times c$ parameters) input for further reducing the parameter size while sharing the available knowledge in spirit of multi-task learning [111]. The second layer (0 parameter) performs a *tanh* scaling (the range of $[-1,1]$) to convert the region position parameters into the percentage so as to allow for positioning individual regions outside of the input image boundary. This specially takes into account the cases that only partial person is detected sometimes.

Note that, unlike the soft attention maps applied to the input feature representation $\boldsymbol{X}^l$, the hard regional attention is enforced on that of the corresponding network block to generate T



(a) STN [82]

(b) HAN Hard Attention

Figure 3.3: Schematic comparison between (a) Spatial Transformer Network (STN) [82] and (b) HAN Hard Attention. Global feature and hard attention are jointly learned in a multi-task design. "$\boldsymbol{H}$": Hard attention module; "$\boldsymbol{F_g}$": Global feature module; "$\boldsymbol{F_l}$": Local feature module.

different parts which are subsequently fed into the corresponding streams of the *local* branch (see the dashed arrow on the top of Fig 3.1).

Remarks. The proposed hard attention modelling is conceptually similar to the Spatial Transformer Network (STN) [82] because both are designed to learn a transformation matrix for discriminative region identification and alignment. However, they differ significantly in design: **(1)** The STN attention is *network-wise* (one level of attention learning) whilst our HA is *module-wise* (multiple levels of attention learning). The latter not only eases the attention modelling complexity (divide-and-conquer design), but also provides additional attention refinement in a sequential manner. **(2)** The STN utilises a separate large sub-network for attention modelling whilst the HAN exploits a much smaller sub-network by sharing the majority model learning with the target-task network using a multi-task learning design (Fig. 3.3), therefore superior in both higher efficiency and lower overfitting risk. **(3)** The STN considers only hard attention whilst HAN models both soft and hard attention in an end-to-end fashion so that additional complementary benefits are exploited.

(III) Cross-Attention Interaction Learning. Given the joint learning of soft and hard attention as above, we further consider a cross-attention interaction mechanism for enriching their joint learning harmony by interacting the *attended* local and global features across branches. Specifically, at the l-th level, we utilise the global-branch feature $\boldsymbol{X}_G^{(l,k)}$ of the k-th region to enrich the corresponding local-branch feature $\boldsymbol{X}_L^{(l,k)}$ by tensor addition as

$$\tilde{\boldsymbol{X}}_L^{(l,k)} = \boldsymbol{X}_L^{(l,k)} + \boldsymbol{X}_G^{(l,k)} \tag{3.8}$$

where $\boldsymbol{X}_G^{(l,k)}$ is computed by applying the hard regional attention of the $(l+1)$-th level's HA attention module (see the dashed arrow in Fig. 3.1). By doing so, we can simultaneously reduce the complexity of the local branch (fewer layers) since the learning capability of the global branch can be partially shared. During model training by back-propagation, the global branch takes gradients from both the global and local branches as

$$\Delta \boldsymbol{W}_G^{(l)} = \frac{\partial \mathcal{L}_G}{\partial \boldsymbol{X}_G^{(l)}} \frac{\partial \boldsymbol{X}_G^{(l)}}{\partial \boldsymbol{W}_G^{(l)}} + \sum_{k=1}^{T} \frac{\partial \mathcal{L}_L}{\partial \tilde{\boldsymbol{X}}_L^{(l,k)}} \frac{\partial \tilde{\boldsymbol{X}}_L^{(l,k)}}{\partial \boldsymbol{W}_G^{(l)}} \tag{3.9}$$

So, the global $\mathcal{L}_G$ and local $\mathcal{L}_L$ loss quantities concurrently function in optimising the parameters $\boldsymbol{W}_G^{(l)}$ of the global branch. As such, the learning of the global branch is interacted with that of the local branch at multiple levels, whilst both are subject to the same re-id optimisation constraint.

Remarks. By design, cross-attention interaction learning is subsequent to and complementary with the harmonious attention joint reasoning above. Specifically, the latter learns soft and hard attention from the same input feature representations to maximise their compatibility (*joint attention generation*), whilst the former optimises the correlated complementary information between attention refined global and local features under the person re-id matching constraint (*joint attention application*). Hence, the composition of both forms a complete process of joint optimisation of attention selection for person re-id.

Conceptually, our Harmonious Attention (HA) is a principled union of hard *regional* attention [82], soft *spatial* [28] and *channel* attention [110]. This simulates functionally the dorsal and ventral attention mechanism of human brain [112] in the sense of modelling soft and hard attention simultaneously. Soft attention learning aims at selecting *fine-grained* important pixels, whilst hard attention learning at searching *coarse* latent (weakly supervised) discriminative regions. They are thus largely complementary with high compatibility to each other in functionality. Intuitively, their combination and interaction can relieve the modelling burden of challenging soft attention learning, resulting in more discriminative and efficient models.

3.2.3 HAN Model Instantiation

To instantiate HAN models, we build up on the state-fo-the-art computationally efficient depthwise separable conv units [113] in the main implementation[1]. In particular, we use 9 depthwise separable conv units to build the global branch, and 3 for each local stream. We set $T = 4$ regions for hard attention, e.g. a total of 4 local streams. We consider a 3-level attention hierarchy design (Fig. 3.1). The global branch network ends with a *global average pooling* layer and a

[1] Besides, we also consider other building block designs to evaluate the generalisation of the proposed method in our experiments (Table 3.7).

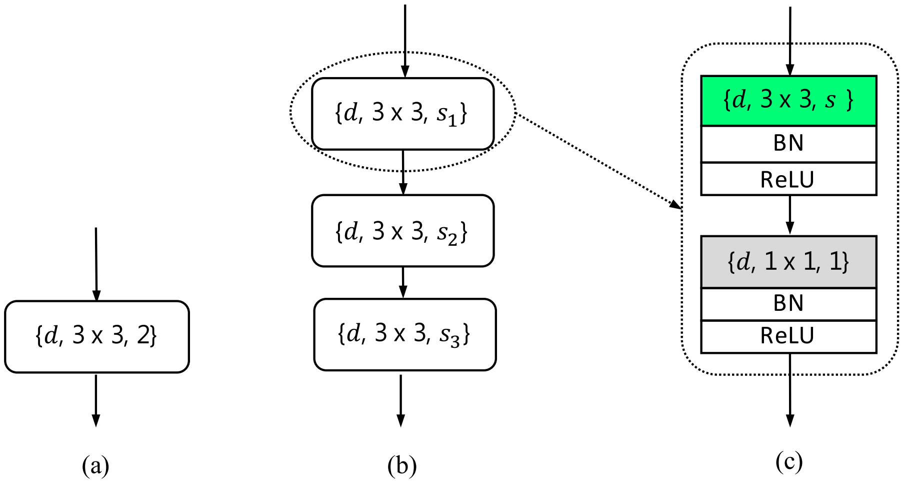


Figure 3.4: Structure of **(a)** local block and **(b)** global block. Each block **(c)** consists of two conv layers. Layer type is indicated by background colour: **grey** for *normal conv*, and **green** for *depthwise separable conv* layers. The three items in the bracket of a conv layer are: filter number, filter shape, and stride.

fully-connected (FC) feature layer with 512 outputs. For the local branch, we also use a 512-D FC feature layer which fuses the global average pooling outputs of all local streams.

To provide diverse options in model efficiency, we explore three HAN models with different inference computational costs. We realise this through varying the stride parameter s of the building block units in the relatively heavier global branch (Fig. 3.4(b)). More specifically, the computational cost (FLOPs) of a HAN model is largely determined by the size of input feature maps per conv layer in each block unit. The stride parameter controls the shifting step size the conv filters travel across the input feature maps, therefore the size of output feature maps and the computational cost of subsequent conv layers. Given the context of hierarchical CNN structure, larger stride values at earlier layer lead to smaller feature maps and lower FLOPs. In our designs, we adopt two strides $\{1, 2\}$ and manage the overall computational complexity of HAN models by positioning the larger stride “2” to different layers. That is, placing the stride “2” to earlier layers yields HANs with higher computational costs, and vice verse. The configurations of the three stride parameters in a global black are summarised in Table 3.1. We will evaluate all these HAN models in our experiments.

3.2.4 Scalable Person Re-ID by HAN

Given a trained HAN model, we obtain a 1,024-D joint feature vector (deep feature representation) by concatenating the local (512-D) and the global (512-D) branch feature vectors. For

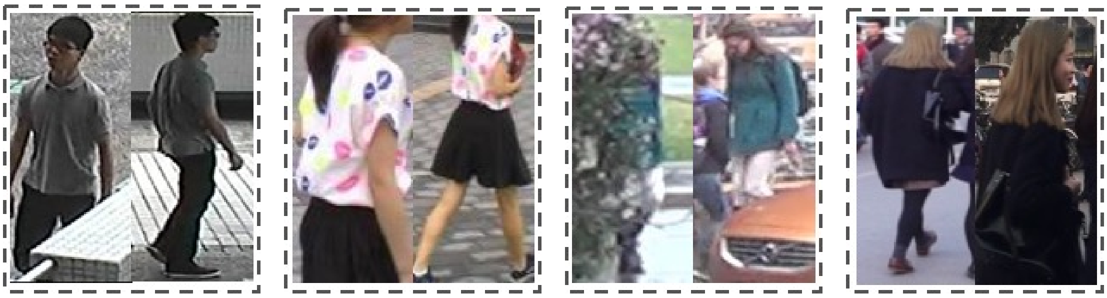
Model	s_1	s_2	s_3	FLOPs
HAN(Small)	2	1	1	3.68×10^8
HAN(Medium)	1	2	1	5.33×10^8
HAN(Large)	1	1	2	7.01×10^8

Table 3.1: The stride configuration for the building block units in the global branch of three varying-efficient HAN models.

person re-id, we deploy this 1,024-D deep feature representation using *only* a generic distance metric *without* any camera-pair specific distance metric learning, e.g. the L2 distance.

Specifically, given a test probe image $\boldsymbol{I}^p$ from one camera view and a set of test gallery images $\{\boldsymbol{I}_i^g\}$ from other non-overlapping camera views: (1) We first compute the corresponding 1,024-D feature representation vectors by forward-feeding the images to a trained HAN model, denoted as $\boldsymbol{x}^p = [\boldsymbol{x}_g^p; \boldsymbol{x}_l^p]$ and $\{\boldsymbol{x}_i^g = [\boldsymbol{x}_g^g; \boldsymbol{x}_l^g]\}$. (2) We then apply L2 normalisation on the global and local features, respectively. (3) Lastly, we compute the cross-camera matching distances between $\boldsymbol{x}^p$ and $\boldsymbol{x}_i^g$ by the L2 distance. We rank all gallery images in ascendant order by the L2 distances against the probe image. The percentage of true matches of the probe person image in top ranks indicate the goodness of the learned HAN deep features for person re-id matching.

3.3 Experiments



(a) CUHK03 (b) Market-1501 (c) DukeMTMC (d) MSMT17

Figure 3.5: Example cross-view matched person image pairs randomly selected from four person re-id benchmark datasets.

Datasets and Evaluation Protocol. For evaluation, we selected four large-scale re-id benchmark datasets: Market-1501 [40], DukeMTMC [42], CUHK03 [41], and MSMT17 [39]. Figure 4.4 shows example person bounding box images. To make a fair comparison against existing methods, we adopted the standard person re-id evaluation setting including the training and testing ID split as summarised in Table 4.1. For DukeMTMC, we followed the person re-id evaluation setup as [61]. These datasets present diverse re-id test scenarios with varying training and testing scales under realistic viewing conditions exposed to large variations in human pose and strong similarities among different people, therefore enabling extensive model evaluations in both model learning and generalisation capabilities. We also considered the model performance with re-ranking [114] as a post-processing. To measure re-id accuracy performance, we used the cumulative matching characteristic (CMC) and mean Average Precision (mAP) metrics. For model efficiency measure, we used the FLoating-point OPerations (FLOPs), i.e. the number of multiply-adds, consumed in forwarding a person image through the testing network.

Dataset	# ID	# Train	# Test	# Image	# Cam
CUHK03	1,467	767	700	28,192	2
Market-1501	1,501	751	750	32,668	6
DukeMTMC	1,404	702	702	36,411	8
MSMT17	4,101	1,041	3,060	126,441	15

Table 3.2: Data statistics of person re-id datasets.

Implementation Details. We implemented our HAN model in the Tensorflow framework [115]. All person images were resized to 160×64, unless stated otherwise. For all HAN models, we set the width of building block units at the $1^{\text{st}}/2^{\text{nd}}/3^{\text{rd}}$ levels as: $d_1 = 128$, $d_2 = 256$ and $d_3 = 384$ (Fig. 3.1). In each stream, we fixed the size of three levels of hard attention as 24×28, 12×14 and 6×7. For model training, we used SGD algorithm at the initial learning rate 3×10^{-2} with momentum of 0.9, learning rate decay of 0.1, and weight decay of 0.0005. We set the batch size to 32 and the epoch number to 300 with learning rate decayed at epochs of 100, 200, and 250. To enable efficient and scalable model training, we did *not* adopt any data argumentation methods (e.g. scaling, rotation, flipping, and colour distortion), *neither* ImageNet model pre-training. Existing deep re-id methods typically benefit significantly from these operations, suffering much higher computational cost, notoriously difficult and time-consuming model tuning, and the implicit undesired dependence on the source data.

3.3.1 Further Analysis and Discussions

To provide more detailed examinations and insights, we conducted a sequence of component analysis using HAN (Large) on Market-1501 and DukeMTMC in the Single-Query setting.

Joint Local and Global Features. We evaluated the effect of joint local and global features by comparing their individual re-id performances against that of the joint feature. Table 3.3 shows

Dataset	Market-1501		DukeMTMC	
Metric (%)	R1	mAP	R1	mAP
Global	88.9	72.2	77.9	59.8
Local	89.3	73.0	77.8	59.8
Global+Local	**91.6**	**76.7**	**80.6**	**64.1**

Table 3.3: Evaluating the global and local-level features.

that: **(1)** Either feature representation *alone* is already very discriminative for person re-id. **(2)** A further performance gain is obtained by joining the two representations, yielding 2.7%(91.6-88.9) in Rank-1 boost and 4.5%(76.7-72.2) in mAP increase on Market-1501. Similar trends are observed on DukeMTMC (Table 3.9). These validate the complementary effect of jointly learning local and global features in the harmonious attention context by our HAN model.

Dataset	Market-1501		DukeMTMC	
Metric (%)	R1	mAP	R1	mAP
No Attention	85.2	64.2	72.8	50.7
SSA	86.3	65.3	74.6	51.2
SCA	87.2	67.7	74.8	53.0
SSA+SCA	88.9	69.9	77.2	56.1
HRA	87.9	70.3	77.1	60.0
SSA+HRA	89.5	72.1	77.5	60.1
SCA+HRA	90.1	74.9	78.9	62.9
HAN(All)	**91.6**	**76.7**	**80.6**	**64.1**

Table 3.4: Comparative evaluation of individual types of attention in our HA model. SSA: Soft Spatial Attention; SCA: Soft Channel Attention; HRA: Hard Regional Attention.

Different Types of Attention. We tested the effect of individual attention components in the HAN model: Soft Spatial Attention (SSA), Soft Channel Attention (SCA), and Hard Regional Attention (HRA). Table 3.4 shows that: **(1)** Each type of attention *in isolation* brings person re-id performance gain; **(2)** SSA+SCA gives a further accuracy boost, suggesting the complementary information between the two soft attention discovered by our model; **(3)** When combining the hard and soft attention (SSA, SCA, or both), the model performance can be further improved. This indicates that our method is effective in identifying and exploiting the complementary benefits between coarse-grained hard attention and fine-grained soft attention.

Cross-Attention Interaction Learning. We evaluated the benefit of cross-attention interaction learning (CAIL) between the global and local branches. Table 3.5 shows three observations: (1) CAIL benefits clearly the learning of global feature, local feature and their combination. (2) The local branch obtains substantially more performance gain, which is expected since its

Dataset	Market-1501		DukeMTMC	
Metric (%)	R1	mAP	R1	mAP
Global (**w/o** CAIL)	84.1	64.0	73.2	53.3
Global (**w/** CAIL)	**88.9**	**72.2**	**77.9**	**59.8**
Local (**w/o** CAIL)	17.1	7.6	24.3	14.3
Local (**w/** CAIL)	**89.3**	**73.0**	**77.8**	**59.8**
Global+Local **w/o** CAIL	72.1	50.9	56.7	40.1
Global+Local (**w/** CAIL)	**91.6**	**76.7**	**80.6**	**64.1**

Table 3.5: Evaluating the Cross-Attention Interaction Learning (CAIL) component.

design is of super-lightweight with insufficient learning capacity on its own; With CAIL, it also simultaneously borrows the representation learning capacity from the global branch. This overall design aims for minimising the model parameter redundancy. (3) Without CAIL, the combined feature is even inferior to the global feature alone, due to the negative impact from the very weak and incompatible local feature. This suggests that CAIL also plays a significant bridging role between the two branches in our model formulation.

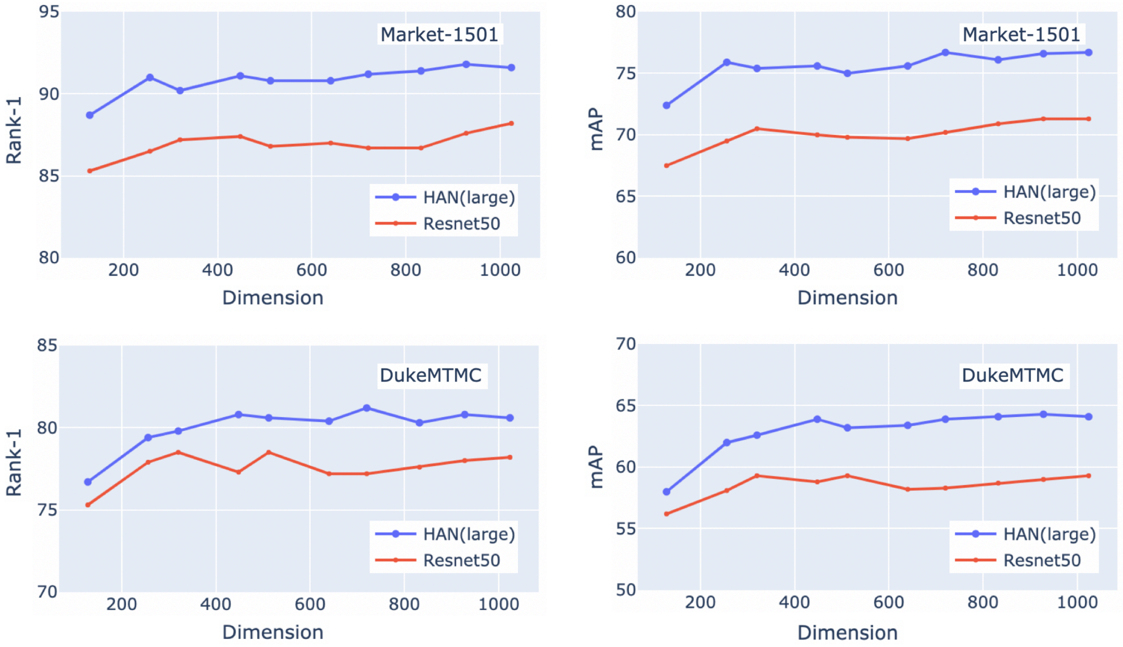


Figure 3.6: Evaluating a set of feature dimensions $\{128, 256, 320, 448, 512, 640, 720, 832, 928, 1024\}$ w.r.t. the re-id performance on Market-1501 and DukeMTMC.

Objective Loss Function. We evaluated the choice of objective loss function in HAN. In particular, we additionally tested the common triplet ranking loss. To more effectively and efficiently impose useful triplet constraints, we exploited the online triplet selection strategy in a hard sample mining principle [116, 117].

Dataset	Market-1501		DukeMTMC	
Metric (%)	R1	mAP	R1	mAP
Cross-Entropy	**91.6**	**76.7**	**80.6**	**64.1**
Triplet	63.7	41.7	57.8	37.6
Cross-Entropy + Triplet	91.5	76.3	79.9	61.5

Table 3.6: Evaluating different types of objective functions.

Dataset	Market-1501		DukeMTMC		FLOPs
Metric (%)	R1	mAP	R1	mAP	
Depthwise	**91.6**	**76.7**	**80.6**	**64.1**	$\mathbf{7.01 \times 10^8}$
Inception	91.2	75.7	80.5	63.8	1.09×10^9
Residual	91.0	75.0	78.5	62.1	1.34×10^9

Table 3.7: Evaluating different types of building blocks.

Specifically, for each mini-batch training we identify on-the-fly and use only those hard triplets yielding positive loss values while throwing away the remaining ones that fulfil the triplet constraints with zero loss values.

We have some interesting observations in Table 3.6: (1) Using the triplet loss *alone* in the tiny HAN model gives significantly inferior re-id performance. The plausible reason is that such pairwise comparison based objective has an unnoticed need for large (less efficient though more expressive) neural network models. (2) Combining the triplet and cross-entropy loss functions can only achieve similar results as using the latter alone. This suggests that the triplet ranking loss is hardly able to provide complementary supervision information, due to the strong capability of identifying re-id attention by the HAN. This favourably eliminates the need of detecting subtle discrepancy between visually similar different persons through exhuastive (a quadratic number of identity pairs) pairwise contrasts in the triplet loss. (3) With a strong attention model like HAN, it is likely that the simpler cross-entropy loss suffices to induce a discriminative person re-id model.

Network Building Blocks. We examined the generalisation capability of HAN in the incorporation of three state-of-the-art CNN building block designs: (1) Inception-A/B unit [47, 43], (2) Residual bottleneck unit [3], and (3) Depthwise separable conv unit used in our main models [113]. Table 3.7 shows that HAN is able to effectively accommodate varying *types* of conv building blocks. Among the three designs, it is interestingly found that the depthwise one is the most cost-effective unit, yielding both the best accuracy and efficiency. This provides an unusual example that the lightweight depthwise conv units are *not necessarily* inferior in recognition

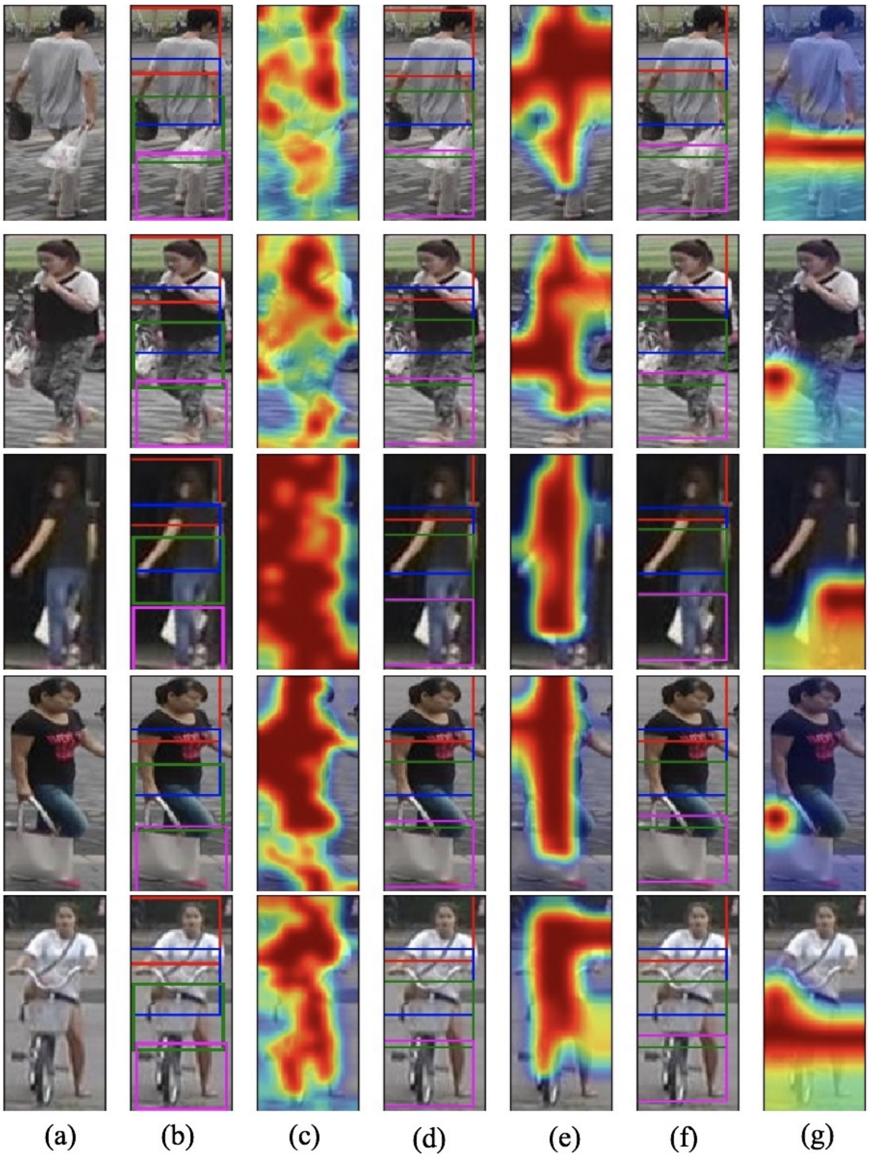


Figure 3.7: Visualisation of the harmonious re-id attention. From left to right, **(a)** the original image, **(b)** the 1st-level of hard attention, **(c)** the 1st-level of soft attention, **(d)** the 2nd-level of hard attention, **(e)** the 2nd-level of soft attention, **(f)** the 3rd-level of hard attention, **(g)** the 3rd-level of soft attention.

performance, contrary to the existing finding in coarse-grained object detection and recognition tasks [62].

Feature Dimension. In addition to feature extraction cost, feature dimension is another scalability factor in the large scale re-id search process, regarding to data transportation, storage size, and matching speed. We evaluated this factor by comparing our method with the golden standard model ResNet50, using a set of feature dimensions ranging from 128 to 1024. As shown in Fig. 3.6, the HAN(large) model not only delivers consistent performance advantage over all the feature dimensions, but also enables the use of lower-dimensional feature vectors whilst simultaneously yielding a similar or even better re-id performance. This verifies the superior scalability of our method in terms of memory usage and matching efficiency, therefore scalable to small feature vectors for better data transportation and potential deployment in the cloud or at the edge. The margin gets smaller on the more challenging DukeMTMC dataset when using very low feature dimensions (e.g. 128) due to too limited feature representation capacity to fully exploit the learning capability of HAN.

Visualisation of Harmonious Attention. We visualised both learned soft attention and hard attention discovered by the HAN re-id model at three different network levels. Figure 3.7 shows that: **(1)** Hard attention localises four body parts well at all three levels, approximately corresponding to head+shoulder (red), upper-body (blue), upper-leg (green) and lower-leg (violet). **(2)** Soft attention focuses on the discriminative pixel-wise selections progressively in spatial localisation, e.g. attending hierarchically from the global whole body by the 1$^{\text{st}}$-level spatial soft attention (c) to local salient parts (e.g. object associations) by the 3$^{\text{rd}}$-level spatial soft attention (g). This shows compellingly the effectiveness and collaboration of soft and hard attention joint learning.

3.3.2 Comparing State-of-the-Art Re-ID Methods

Evaluation on Market-1501. We evaluated the HAN models in comparison to recent state-of-the-art methods on the Market-1501 dataset. Table 3.8 shows clear superiority and advantages of the proposed HAN in model cost-effectiveness. Specifically, in the standard model training setting, G-SCNN [104] is featured with the best FLOPs, but far inferior to many alternative methods including all HAN models in terms of re-id performance. HAN(Small) is on par with the recent art MLFN [126] in re-id matching accuracy whilst simultaneously achieving an efficiency advantage of 7× cheaper inference.

With a re-ranking based post-processing, re-id models generally can further improve the accuracy performance. Note, this benefit comes with a higher computational cost, e.g. multiple times standard search expense. Interestingly, the fastest model HAN(Small) benefits the most, achieving superior model efficiency and discrimination simultaneously against other existing alternative methods.

As a training data augmentation strategy, random erasing is shown to be effective for improving re-id model generalisation. For example, the strong models GCS [102] and SGGNN [130]

Query Type	SQ		MQ		FLOPs
Metric (%)	R1	mAP	R1	mAP	
CRAFT [119]	68.7	42.3	77.0	50.3	N/A
CAN [120]	60.3	35.9	72.1	47.9	$>1.55\times10^{10}$
G-SCNN [104]	65.8	39.5	76.0	48.4	$\approx\mathbf{1.11}\times\mathbf{10}^{8}$
HPN [83]	76.9	-	-	-	$\approx1.82\times10^{10}$
SVDNet [121]	82.3	62.1	-	-	$>3.80\times10^{9}$
MSCAN [74]	80.3	57.5	86.8	66.7	1.36×10^{9}
DLPA [75]	81.0	63.4	-	-	$>7.29\times10^{8}$
PDC [76]	84.1	63.4	-	-	$\gg9.82\times10^{9}$
GLAD [122]	89.9	73.9	-	-	$\gg7.99\times10^{9}$
DPFL [50]	88.9	73.1	92.3	80.7	$\approx1.2\times10^{10}$
AACN [78]	85.9	66.9	89.8	75.1	$>1.57\times10^{9}$
DML [123]	89.3	70.5	92.8	79.0	5.69×10^{8}
DaRe-D201[69]	86.0	69.9	-	-	$>4.00\times10^{9}$
PT-D169 [124]	87.7	68.9	-	-	$>3.00\times10^{9}$
AOS [125]	86.5	70.4	91.3	78.3	$\approx3.80\times10^{9}$
BraidNet [103]	83.7	69.5	-	-	$>2.26\times10^{9}$
MLFN [126]	90.0	74.3	92.3	82.4	$\approx2.60\times10^{9}$
CamStyle [127]	88.1	68.7	-	-	$\approx3.80\times0^{9}$
PSE [128]	87.7	69.0	-	-	$>3.80\times10^{9}$
KPM [101]	90.1	75.3	-	-	$>3.80\times10^{9}$
PoseNorm [80]	89.4	72.6	92.9	80.2	$>3.80\times10^{9}$
HAP2S [129]	84.6	69.4	90.2	76.8	$\approx3.80\times10^{9}$
HAN(Small) (Ours)	89.4	73.2	93.2	80.8	3.68×10^{8}
HAN(Medium) (Ours)	90.7	74.5	93.9	81.9	5.33×10^{8}
HAN(Large) (Ours)	**91.6**	**76.7**	**94.2**	**83.4**	7.01×10^{8}
AACN$^{\text{RR}}$[78]	88.7	83.0	92.2	87.3	$>1.57\times10^{9}$
DaRe-D201$^{\text{RR}}$[69]	88.6	82.2	-	-	$>4.00\times10^{9}$
MGCAM$^{\text{RR}}$[52]	83.8	74.3	-	-	3.76×10^{8}
AOS$^{\text{RR}}$[125]	88.7	83.3	92.5	88.6	$\approx3.80\times10^{9}$
PSE$^{\text{RR}\dagger}$ [128]	90.3	84.0	-	-	$>3.80\times10^{9}$
HAN(Small)$^{\text{RR}}$	91.2	85.5	93.7	90.1	$\mathbf{3.68}\times\mathbf{10}^{8}$
HAN(Medium)$^{\text{RR}}$	92.0	86.9	94.1	90.8	5.33×10^{8}
HAN(Large)$^{\text{RR}}$	**92.0**	**87.5**	**94.3**	**90.9**	7.01×10^{8}
SGGNN$^{\text{RE}}$ [130]	92.3	**82.8**	-	-	$>3.80\times10^{9}$
GCS$^{\text{RE}}$ [102]	**93.5**	81.6	-	-	$>3.80\times10^{9}$
HAN(Small)$^{\text{RE}}$	90.0	75.3	93.2	82.3	$\mathbf{3.68}\times\mathbf{10}^{8}$
HAN(Medium)$^{\text{RE}}$	90.9	77.9	93.7	84.0	5.33×10^{8}
HAN(Large)$^{\text{RE}}$	91.1	78.1	94.1	84.7	7.01×10^{8}
RW$^{\text{RR}\dagger,\text{RE}}$ [101]	92.7	82.5	-	-	$>3.80\times10^{9}$
HAN(Small)$^{\text{RR,RE}}$	92.3	87.5	93.8	91.0	$\mathbf{3.68}\times\mathbf{10}^{8}$
HAN(Medium)$^{\text{RR,RE}}$	92.9	88.8	94.5	92.0	5.33×10^{8}
HAN(Large)$^{\text{RR,RE}}$	**93.1**	**89.6**	**94.8**	**92.5**	7.01×10^{8}

Table 3.8: Performance evaluation on **Market-1501**. “$^{\text{RR}}$”: Using the re-ranking method in [114]. “†”: Using a newly proposed re-ranking method. “$^{\text{RE}}$”: Using random erasing data augmentation [118]. D201: DenseNet201; D169: DenseNet169; R50: ResNet50. SQ: Single-Query; MQ: Multi-Query.

benefit significantly, achieving the best re-id matching rates. However, this model is also largely expensive in the computational cost, e.g. more than 10× cost of HAN(Small).

When applying both random easing and re-ranking, a complementary benefit is likely to be obtained. In this setting, our HAN(Small) suffices to outperform the competitor RW [101] in both accuracy performance and inference cost. If more computational budge is allowed, we can further improve the model performance by deploying HAN(Large).

Evaluation on DukeMTMC. We compared the HAN models with recent state-of-the-art re-id methods on the DukeMTMC dataset. Compared to Market-1501, this benchmark provides a similar training and testing data scale, but the person images have more variations in resolution and background clutter. This is due to wider camera views and more complex scene layout, therefore presenting a more challenging re-id task.

Table 3.9 shows that all HAN methods again present superior model cost-effectiveness as compared to alternative state-of-the-art methods. Overall, we have similar comparative observations as on Market-1501. In the standard training setting, our HAN models are the most efficient solutions whilst achieving top performances. With re-ranking, PSE [128] slightly outperforms HAN models with up to 10× more expensive in the computational cost. Similar contrasts between HAN and the competitors are observed when using random erasing based data augmentation alone or along with re-ranking.

Evaluation on CUHK03. We evaluated the HAN models on both manually labelled and auto-detected (more severe misalignment) person bounding boxes on the CUHK03 benchmark. We adopted the 767/700 identity split rather than 1367/100 since the former defines a more realistic and challenging re-id task. In the standard setting, the training set is rather small, with only 7,365 training images vs 12,936 and 16,522 on Market-1501 and DukeMTMC. This generally imposes an extreme challenge to the training of deep models, particularly in case of using *no* large auxiliary data (e.g. ImageNet) for model pre-training like our HAN models.

Table 3.10 shows that the HAN models still achieve competitive re-id matching accuracy, although outperformed by two recent computationally expensive approaches MLFN [126] and DaRe-R50 [69] which benefit substantially from ImageNet in model pre-training. Among all competitors, our models are most efficient therefore more scalable to large scale re-id deployments in practical use. If more computational resource is available, re-ranking can be applied for all methods to further improve the re-id performance.

Metric (%)	R1	mAP	FLOPs
LSRO-R50 [61]	67.7	47.1	$>3.80\times10^{9}$
SVDNet-R50 [121]	76.7	56.8	$>3.80\times10^{9}$
DPFL [50]	79.2	60.6	$\approx1.2\times10^{10}$
AACN [78]	76.8	59.3	$>1.57\times10^{9}$
PT-R50 [124]	78.5	56.9	$>3.80\times10^{9}$
DaRe-R50 [69]	75.2	57.4	$>3.80\times10^{9}$
BraidNet [103]	76.4	59.5	$>2.26\times10^{9}$
MLFN [126]	**81.0**	62.8	$\approx2.60\times10^{9}$
CamStyle [127]	75.3	53.5	$\approx3.80\times10^{9}$
AOS [125]	79.2	62.1	$\approx3.80\times10^{9}$
PSE [128]	79.8	62.0	$>3.80\times10^{9}$
KPM [101]	80.3	63.2	$>3.80\times10^{9}$
PoseNorm [80]	73.6	53.2	$>3.80\times10^{9}$
HAP2S [129]	75.9	60.6	$\approx3.80\times10^{9}$
HAN(Small)	78.9	61.9	$\mathbf{3.68\times10^{8}}$
HAN(Medium)	80.0	63.4	5.33×10^{8}
HAN(Large)	80.6	**64.1**	7.01×10^{8}
DaRe-R50^{RR}[69]	80.4	74.5	$>3.80\times10^{9}$
AOS^{RR} [125]	84.1	78.2	$\approx3.80\times10^{9}$
PSE^{RR†} [128]	**85.2**	**79.8**	$>3.80\times10^{9}$
HAN(Small)^{RR}	83.2	77.9	$\mathbf{3.68\times10^{8}}$
HAN(Medium)^{RR}	84.0	78.8	5.33×10^{8}
HAN(Large)^{RR}	84.0	79.5	7.01×10^{8}
SGGNN^{RE} [130]	81.1	68.2	$>3.80\times10^{9}$
GCS^{RE} [102]	**84.9**	**69.5**	$>3.80\times10^{9}$
HAN(Small)^{RE}	79.4	64.0	$\mathbf{3.68\times10^{8}}$
HAN(Medium)^{RE}	80.5	64.7	5.33×10^{8}
HAN(Large)^{RE}	80.7	65.9	7.01×10^{8}
RW^{RR†,RE} [101]	80.7	**82.5**	$>3.80\times10^{9}$
HAN(Small)^{RR,RE}	83.9	79.6	$\mathbf{3.68\times10^{8}}$
HAN(Medium)^{RR,RE}	84.2	80.2	5.33×10^{8}
HAN(Large)^{RR,RE}	**84.6**	81.3	7.01×10^{8}

Table 3.9: Performance evaluation on **DukeMTMC**. “^{RR}”: Using the re-ranking method in [114]. “†”: Using a newly proposed re-ranking method. “^{RE}”: Using random erasing data augmentation [118]. R50: ResNet50; D201: DenseNet201.

Metric (%)	Labelled		Detected		FLOPs
	R1	mAP	R1	mAP	
IDE-C [114]	15.6	14.9	15.1	14.2	7.25×10^{8}
XQDA-C [114]	21.9	20.0	21.1	19.0	7.25×10^{8}
IDE-R50 [114]	22.2	21.0	21.3	19.7	3.80×10^{9}
XQDA-R50[114]	32.0	29.6	31.1	28.2	3.80×10^{9}
SVDNet-C [121]	-	-	27.7	24.9	$>7.25\times10^{8}$
SVDNet-R50 [121]	-	-	41.5	37.3	$>3.80\times10^{9}$
DPFL [50]	43.0	40.5	40.7	37.0	$\approx1.2\times10^{10}$
PT-R50 [124]	45.1	42.0	41.6	38.7	$>3.80\times10^{9}$
AOS [125]	-	-	47.1	43.3	$\approx3.80\times10^{9}$
DaRe-R50 [69]	**58.1**	**53.7**	**55.1**	**51.3**	$>3.80\times10^{9}$
MLFN [126]	54.7	49.2	52.8	47.8	$>2.26\times10^{9}$
HAN(Small)	42.7	42.4	40.9	40.0	**3.68×10^{8}**
HAN(Medium)	42.0	42.3	42.8	42.0	5.33×10^{8}
HAN(Large)	46.5	46.1	47.5	45.5	7.01×10^{8}
AOS$^{\text{RR}}$ [125]	-	-	54.6	56.1	$\approx3.80\times10^{9}$
MGCAM$^{\text{RR}}$ [52]	50.1	50.2	46.7	46.9	3.76×10^{8}
DaRe-R50$^{\text{RR}}$[69]	**66.0**	**66.7**	**62.8**	**63.6**	$>3.80\times10^{9}$
HAN(Small)$^{\text{RR}}$	49.6	54.3	46.9	51.6	**3.68×10^{8}**
HAN(Medium)$^{\text{RR}}$	50.1	55.2	49.7	54.0	5.33×10^{8}
HAN(Large)$^{\text{RR}}$	53.6	58.7	54.9	57.9	7.01×10^{8}

Table 3.10: Performance evaluation on **CUHK03**. "$^{\text{RR}}$": Using the re-ranking method in [114]. C: CaffeNet; R50: ResNet50; D201: DenseNet201.

Metric (%)	R1	mAP	FLOPs
GoogLeNet[5]	47.6	23.0	1.57×10^{9}
PDC [76]	58.0	29.7	$\gg9.82\times10^{9}$
GLAD [122]	**61.4**	**34.0**	$\gg7.99\times10^{9}$
ResNet50 [3]	59.7	33.7	3.80×10^{9}
HAN(Small)	56.3	29.2	**3.68×10^{8}**
HAN(Medium)	57.1	30.3	5.33×10^{8}
HAN(Large)	60.1	32.6	7.01×10^{8}
ResNet50$^{\text{RR}}$ [3]	64.6	**47.6**	3.80×10^{9}
HAN(Small)$^{\text{RR}}$	61.8	41.8	**3.68×10^{8}**
HAN(Medium)$^{\text{RR}}$	63.8	42.9	5.33×10^{8}
HAN(Large)$^{\text{RR}}$	**66.2**	46.2	7.01×10^{8}

Table 3.11: Performance evaluation on **MSMT17**. "$^{\text{RR}}$": Using the re-ranking method [114].

Evaluation on MSMT17. We evaluated the HAN models on the new large scale MSMT17 benchmark tested by only a few methods. Having more training data typically benefits larger neural networks due to a reduced model fitting risk, and lightweight networks may be therefore less competitive in accuracy due to relatively inferior representative capabilities. This facilitates a more extensive test on both model learning capacity and generalisation of our lightweight HAN against existing more elaborative and "heavier" deep re-id models, given the larger training and testing sets in terms of the number of images, identities, and cameras. This test is not only

Dataset	Market-1501 (SQ)		Market-1501 (MQ)		DukeMTMC		CUHK03 (L)		CUHK03 (D)		MSMT17		FLOPs
Metric (%)	R1	mAP	R1	mAP	R1	mAP	R1	mAP	R1	mAP	R1	mAP	
MobileNet	84.6	64.3	89.3	72.8	72.1	50.9	36.1	35.5	35.0	34.3	53.8	26.4	5.69×10^8
ShuffleNet	80.6	58.4	86.4	67.0	70.0	48.2	35.6	35.1	33.2	33.3	42.8	18.9	$\mathbf{1.40\times10^8}$
CondenseNet	83.5	62.8	88.2	70.8	72.6	51.3	33.4	33.5	32.1	31.8	54.2	26.5	2.74×10^8
HAN(Small)	89.4	73.2	93.2	80.8	78.9	61.9	42.7	42.4	40.9	40.0	56.3	29.2	3.68×10^8
HAN(Medium)	90.7	74.5	93.9	81.9	80.0	63.4	42.0	42.3	42.8	42.0	57.1	30.3	5.33×10^8
HAN(Large)	**91.6**	**76.7**	**94.2**	**83.4**	**80.6**	**64.1**	**46.5**	**46.1**	**47.5**	**45.5**	**60.1**	**32.6**	7.01×10^8

Table 3.12: Comparisons with efficient neural networks. SQ: Single-Query; MQ: Multi-Query; L: Labelled: D: Detected.

complementary to the other re-id benchmark tests, but also a good evaluation on small networks like HAN models in order to evaluate the models learning capacity when larger training and test data are given whilst having less parameters.

Table 3.11 shows that the heavy model GLAD [122] achieves the best results in the standard setting, but only *slightly* outperforming the HAN models whilst at over 10× more computational costs. Besides, HAN(Large) matches the accuracy performance of ResNet50 with merely 18% inference cost. These suggest that the cost-effectiveness advantages of our HAN models remain on larger scale re-id learning and deployments, and importantly the absolute performances of HAN models are still competitive. The advantages of HAN are similar in case of using re-ranking. This test broadly examines the ability of neural network models in compromising between model complexity (learning capacity) and computational efficiency (processing speed) often required in large scale re-id deployments.

3.3.3 Comparing State-of-the-Art Efficient Networks

We compared the proposed HAN models with three state-of-the-art compact neural network models: MobileNet [62], ShuffleNet [63], and CondenseNet [64]. These competitors are general-purpose lightweight neural networks therefore directly applicable for person re-id although not evaluated in the original works.

Table 3.12 shows that our HAN models achieve the best performances at competitive inference computational costs. In particular, HAN(Small) significantly outperforms MobileNet whilst enjoying more efficient inference. While the CondenseNet and ShuffleNet are more efficient than HAN, their re-id matching performances are the worst. HAN(Large) further improves the model generalisation capability by extra 3.33×10^8 (7.01-3.68) FLOPs per image. These indicate the cost-effectiveness advantages of the proposed HAN models in person re-id over state-of-the-art efficient network designs.

3.4 Summary

In this chapter, we present a cost-effective Harmonious Attention Network (HAN) framework for joint learning of person re-identification attention selection and feature representations. In contrast to existing re-id deep learning methods that typically ignore the model efficiency issue, the HAN model is designed to scale up large deployments whilst simultaneously achieving top re-id matching performances. This is realised by designing a Harmonious Attention mechanism enabling to establish *lightweight* CNN architectures with sufficient discrimination learning capability. Moreover, we introduce a cross-attention interaction learning strategy to enhance the joint optimisation of attention selection and re-id features. Extensive evaluations have been conducted on four large re-id benchmarks with varying training and test scales to validate the cost-effectiveness advantages of our HAN model over state-of-the-art re-id methods and scalable neural network designs. We also provide a series of detailed model component analysis and insightful discussions on the HAN model cost-effectiveness superiority.

Chapter 4

Scalable Distillation Network Design

4.1 Overview

In this chapter, we investigate the scalability problem for person search. We explore the potential of knowledge distillation [94] by developing a *Hierarchical Distillation Learning* (HDL) method. The core idea behind the HDL is to transfer the person search knowledge of a heavy teacher model that can be optimised more discriminatively with stronger learning capability into a lightweight student model with weaker learning capability. Whilst knowledge distillation has been previously studied mostly in single label image classification [131, 94, 132, 133], it has not been explored for the more complex person search problem with two different tasks involved. To this end, we design a novel approach for distilling comprehensive knowledge in the teacher network hierarchy including feature representation, attention, and prediction. To facilitate distillation, we further develop a strong joint learning teacher model for ensuring the knowledge quality which is lacking in the literature, and a structurally consistent and computationally efficient student model.

4.2 Hierarchical Distillation Learning

For model training, we often collect m training scene images $\mathcal{I} = \{\boldsymbol{I}_i\}_{i=1}^{m}$ captured from multiple camera views. The annotation includes person bounding boxes $\mathcal{Y}_{box}$ and identity labels $\mathcal{Y}_{id} = \{y_i\}_{i=1}^{n}$ on a total of n_{id} training people, i.e. $y_i \in \{1, \cdots, n_{\text{id}}\}$. A single unconstrained scene image may contain multiple (varying) person instances. The objective is to learn an efficient

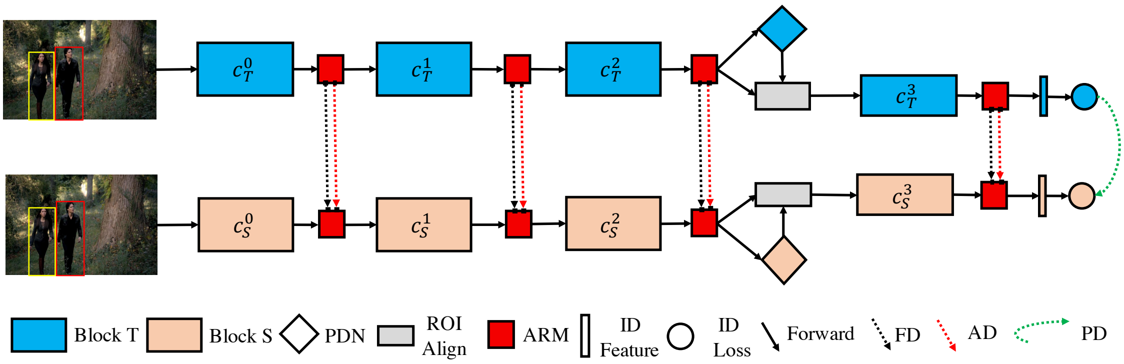


Figure 4.1: Overview of the proposed Hierarchical Distillation Learning (HDL) approach. The HDL process consists of two steps: (1) We first train the heavy teacher model (Sec 4.2.1). See the details in Sec 4.2.1. (2) We then train the lightweight student model (Sec 4.2.2) by knowledge distillation from the teacher model. In test, we deploy the efficient student model for scalable person search. The symbols c_T^j and c_S^j ($j \in \{0,1,2,3\}$) denote channel dimensions in the corresponding j-th block of the teacher and student models. The first layers and standard detection loss functions in both teacher and student models are omitted for simplicity. **T**: Teacher; **S**: Student; **PDN**: Person Detection Network; **ARM**: Attention Residual Module; **FD**: Feature Distillation; **AD**: Attention Distillation; **PD**: Prediction Distillation.

person search model for simultaneous person detection and re-id matching. To this end, we formulate a ***Hierarchical Distillation Learning*** (HDL) approach featured with comprehensive knowledge distillation and joint learning of person detection and person re-id (i.e. person search) in unconstrained surveillance scene imagery data. An architectural overview of the proposed HDL method is depicted in Figure 6.4. In design, the proposed HDL model takes the advantages of knowledge distillation [94]. Specifically, HDL consists of three components: (1) A *teacher* model with a large size and great learning capability, designed to realise a strong person search network (Section 4.2.1). (2) A *student* model with a small size and inferior learning capability, developed for superior inference efficiency in deployment (Section 4.2.2). (3) A *hierarchical distillation* learning strategy, formulated for comprehensive knowledge transfer from the stronger teacher model to the student model (Section 4.2.3). This addresses the hard-to-learn problem in training the small student model. By deploying the student network as the final model in test time, we are able to achieve both superior model generalisation capacity and model inference speed, i.e. higher cost-effectiveness during deployment.

4.2.1 A Strong Joint Learning Teacher Model

By the means of knowledge distillation, the performance of the final (student) model relies heavily on the strength of the teacher model. That is, weaker teacher, weaker student. It is therefore critical and necessary to formulate a strong teacher model. To ease the distillation of person search knowledge, it is also desired that the teacher model can share a similar structure of the student model *functionally* with the ability to jointly conduct both person detection and re-id matching. This avoids distilling the knowledge from two separate teacher networks (one for person detection, one for person re-id) to a single joint leaning student network, which is much more difficult.

Nonetheless, the only existing joint learning teacher model, OIM [84], is significantly inferior to the two-stage followup models [89, 90]. Therefore, using the OIM model as the teacher model will lead to a similarly weak student model. To address this issue, we formulate a *stronger yet simpler* joint learning teacher model.

Teacher Model Architecture. Our teacher model is based on the design idea of Faster R-CNN [32] with person search specific modification. Specifically, it consists of three parts: (i) feature subnet, (ii) person detection subnet, and (iii) person re-id subnet. For design flexibility, any standard deep convolutional networks [134, 44] can be used as the stem network. In the follows, we detail the three parts.

(I) Feature Subnet. To build the feature subnet, we use the lower part of the stem network starting from the first layer to the intermediate layer with $\frac{1}{r}$ down-sampling ratio. This subnet takes as input the scene image $\boldsymbol{I} \in \mathcal{R}^{H \times W \times 3}$ (H and W as image height and width), and outputs the image-level features $\boldsymbol{X}_f \in \mathcal{R}^{\frac{H}{r} \times \frac{W}{r} \times c_f}$ (c_f feature channels). The output features are for both person detection and re-identification tasks simultaneously. This model structure sharing reduces the overall computational costs with only a single unified forward pass needed.

(II) Person Detection Subnet. We subsequently build a person detection subnet (e.g. region proposal net) on top of the output features for detecting candidates in a given scene image. The details are as follows. With a $512 \times 3 \times 3$ conv layer, we first make the features discriminating for person appearance. The followed is the anchor layer for per-feature-location person detection. To make it more effective for person class specifically rather than generic object classes, we use eleven different anchor-box scales and only one aspect ratio ($\frac{h}{w} = 2.44$) as [33]. Finally, we remove the redundant detections by applying a Non-Maximum Suppression process. In training, the person detection is optimised jointly by a softmax cross-entropy classification loss and a spatial location regression loss.

(III) Person Re-Id Subnet. We utilise the rest layers of the stem network to build person re-id subnet. It is based on the outputs of both feature and detection subnets. Specifically, we first use RoIAlign [135] at a spatial scale of 7×7 to crop the detection regions from the output of the feature subnet. This yields the detection-level features $\boldsymbol{X}_p \in \mathcal{R}^{7\times7\times c_f}$. $\boldsymbol{X}_p$ is first processed by batch normalisation, then used as the input of the person re-id subnet to produce the identity discriminative features $\boldsymbol{X}_p^{'} \in \mathcal{R}^{3\times3\times c_p}$, where c_p is the feature dimensions of the last layer in the stem network. To obtain the re-id feature $\boldsymbol{x}_p \in \mathcal{R}^{c_p}$, we globally pool $\boldsymbol{X}_p^{'}$ followed by batch normalisation. In training, we introduce a softmax cross-entropy identity classification loss function for re-id discriminative learning defined as:

$$\mathcal{L}_{\text{ID}} = -\frac{1}{N_{\text{p}}}\sum_{i=1}^{N_{\text{p}}} \log(\bar{\boldsymbol{p}}^i), \tag{4.1}$$

where N_{p} specifies the number of persons detected in the current mini-batch training data. $\bar{\boldsymbol{p}}^i$ is the posterior probability of the i-th training person instance on the ground-truth identity class. Specifically, it is written as:

$$\bar{\boldsymbol{p}}^i = \frac{\exp(\boldsymbol{p}^i)}{\sum_{i\in\mathcal{Y}_{id}} \exp(\boldsymbol{p}^i)}, \tag{4.2}$$

where $\boldsymbol{p}^i$ is the identity class logits predicted by the identity classification layer.

In person search on unconstrained scene images, person detection is often imperfect with inevitable false alarms and misalignment [84]. To mitigate this issue, we further impose a detection refinement loss same as the person detection subnet, in conjugate with the above re-id loss function. This refines the person localisation and suppresses the wrong detections.

Remarks. In this study, we aim for a simple but powerful teacher model. This is in contrast to most existing models that often become more complex making the model analysis and comparison increasingly difficult. For instance, comparison between different models is mostly at the system level therefore less informative. By this simple teacher model we attempt to discourage this trend and answer a question that *how well a simple person search method can perform in a proper design*, which is *unfortunately* lacking in the literature. Interestingly, the proposed teacher model is surprisingly effective although being simple. In comparison, our method has a couple of significant merits: (1) More training friendly; (2) Potentially inspire new research ideas for developing novel joint learning person search models.

4.2.2 An Efficient Joint Learning Student Model

One major weakness of using the standard CNN architecture in the teacher model (e.g. ResNet-50) is the high cost of model inference cost. Whilst facilitating to learn the discriminating feature representations, this is not desirable for large scale deployments. There is hence a need for developing a computationally more efficient student model.

To that end, we design a lightweight building block based on depth-wise separable convolutions, inspired by efficient CNN models such as MobileNets [62, 136]. The details of the student's building block are shown in Fig 4.2. To build the entire student network, we just simply replace all levels of blocks of the teacher network by the proposed efficient blocks. This means that the student model adopts the teacher's overall structure.

Remarks. An important advantage of such a design is that, the teacher and student models are *structurally* consistent. This brings significant convenience for knowledge distillation, as described in the follows.

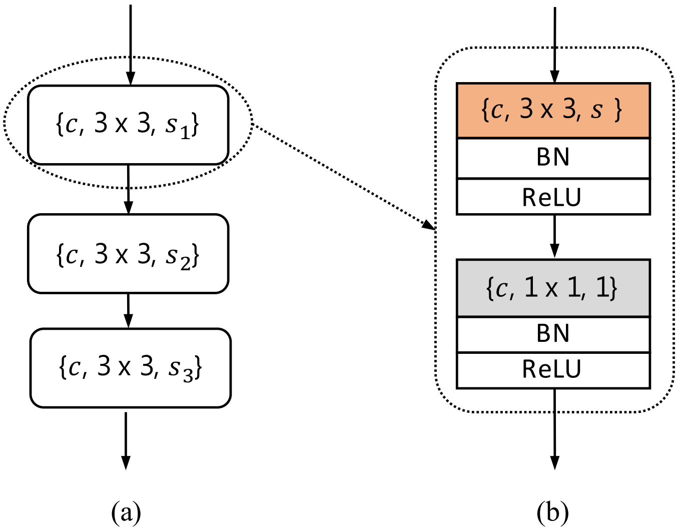


Figure 4.2: **(a)** A student's building block contains three modules. Each module **(b)** in such a block consists of two conv layers. Layer type is indicated by background colour: **grey** for *normal conv*, and **orange** for *depthwise separable conv* layers. The three items in the bracket of a conv layer are: filter number, filter shape, and stride. BN: Batch Normalisation. ReLU: Rectified Linear Unit.

4.2.3 Hierarchical Distillation Learning

Smaller networks are typically inferior for discriminating training. To facilitate the learning of our student model, we propose a hierarchical distillation learning (HDL) strategy that can transfer

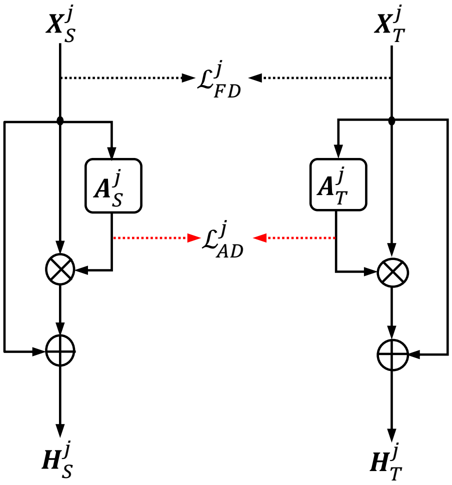


Figure 4.3: Attention residual module in knowledge distillation.

comprehensively the teacher's knowledge for helping the student's training.

Specifically, our HDL method considers three levels of knowledge during distillation: feature, attention, and prediction. For enabling attention distillation, we need an attention learning mechanism for both the teacher and student models. In order to learn and transfer richer attention knowledge distributed across different layers, we consider a *module-wise* attention design. That is, multiple selected building blocks can be attended in a pyramid structure (Fig 6.4). As a side benefit, this may also assist the feature representation learning of both models concurrently.

Attention Residual Module. Formally, the input to an attention module is a 3-D tensor $\boldsymbol{X}^j \in \mathcal{R}^{h\times w\times c}$ where h, w, and c denote the height, width, and channel dimensions, respectively; And j indicates the block level of this module in the entire network. The essence of attention learning is to estimate a salience weight map $\boldsymbol{A}^j \in \mathcal{R}^{h\times w\times c}$ of the same size as $\boldsymbol{X}^j$. In this chapter, we adopt the Attention Residual Module (ARM) design [28] due to its superior learning capability. It is formulated (see Fig 4.3) as:

$$\boldsymbol{H}^j = (1+\boldsymbol{A}^j)*\boldsymbol{X}^j, \tag{4.3}$$

where $\boldsymbol{H}^j \in \mathcal{R}^{h\times w\times c}$ and $\boldsymbol{X}^j \in \mathcal{R}^{h\times w\times c}$ represent the modulated and original features, respectively. To further improve cost-effectiveness, we separate the spatial and channel attention learning as [137, 138].

(I) Feature Distillation. Feature distillation [96] encourages the student to imitate the teacher's representation knowledge. Formally, we denote $\boldsymbol{X}_{S/T}^j$ as the feature maps at the j-th block level of the teacher ($\boldsymbol{X}_T^j$) or student ($\boldsymbol{X}_S^j$) network. For efficiency gain, the student network often has fewer feature channels. As a result, $\boldsymbol{X}_S^j$ and $\boldsymbol{X}_T^j$ are not aligned in channel dimension, which disables channel-to-channel distillation. To address this issue, we consider a 2-D spatial collective distillation scheme by discarding the channel dimension. Specifically, we first accumulate the feature tensor along the channel dimension as:

$$f(\boldsymbol{X}_{S/T}^j) = \sum_i |\boldsymbol{X}_{S/T}^j(\cdot,\cdot,i)|^2, \tag{4.4}$$

where $\boldsymbol{X}_{S/T}^j(\cdot,\cdot,i)$ is the i-th feature channel of $\boldsymbol{X}_{S/T}^j$. We then obtain feature vectors by vectorisation:

$$\boldsymbol{x}_{S/T}^j = vec(f(\boldsymbol{X}_{S/T}^j))$$

We finally design the *feature distillation loss* as:

$$\mathcal{L}_{FD}(\Theta_S) = \frac{1}{2}\sum_{j\in\mathcal{J}} \|\frac{\boldsymbol{x}_S^j}{\|\boldsymbol{x}_S^j\|_2} - \frac{\boldsymbol{x}_T^j}{\|\boldsymbol{x}_T^j\|_2}\|_2 \tag{4.5}$$

where Θ_S denotes the parameters of the student model, and $\mathcal{J}$ the set of all block levels involved.

(II) Attention Distillation. Attention distillation [96] aims for salience knowledge transfer. Specifically, we have the 3-D attention maps $\boldsymbol{A}_S^j$ and $\boldsymbol{A}_T^j$ from the student and teacher models at the j-th level. Similar to feature distillation, we first perform a channel-dimensional accumulation and vectorisation by computing $\boldsymbol{a}_{S/T}^j = vec(f(\boldsymbol{A}_{S/T}^j))$, then formulate the *attention distillation loss* as:

$$\mathcal{L}_{AD}(\Theta_S) = \frac{1}{2}\sum_{j\in\mathcal{J}} \|\frac{\boldsymbol{a}_S^j}{\|\boldsymbol{a}_S^j\|_2} - \frac{\boldsymbol{a}_T^j}{\|\boldsymbol{a}_T^j\|_2}\|_2, \tag{4.6}$$

This essentially constrains the student model to mimic the attending behaviour optimised by the teacher model.

(III) Prediction Distillation. By prediction distillation [94], the student model attempts to simulate the high-level classification actions of the teacher model. Since the class space is the same for both models, their predictions are structurally consistent therefore allowing element-wise alignment. Formally, we design the *prediction distillation loss* as:

$$\mathcal{L}_{PD}(\Theta_S) = t^2 \sum_{i\in\mathcal{Y}_{id}} \tilde{\boldsymbol{p}}_S^i \log\frac{\tilde{\boldsymbol{p}}_S^i}{\tilde{\boldsymbol{p}}_T^i} \tag{4.7}$$

which minimises the Kullback-Leibler divergence between the softened per-identity predictions $\tilde{\boldsymbol{p}}_S^i$ (by student) and $\tilde{\boldsymbol{p}}_T^i$ (by teacher). The temperature parameter t controls the softening degree as:

$$\tilde{\boldsymbol{p}}^i_{S/T} = \frac{\exp\left(\boldsymbol{p}^i_{S/T}/t\right)}{\sum_{i \in \mathcal{Y}_{id}} \exp\left(\boldsymbol{p}^i_{S/T}/t\right)} \tag{4.8}$$

where $\boldsymbol{p}^i_{S/T}$ is the identity class logits predicted by the student or teacher model. As the gradient magnitudes produced by the soft targets $\tilde{\boldsymbol{p}}^i_{S/T}$ are scaled by $\frac{1}{t^2}$, we multiply this loss term by a factor t^2. This is to ensure that the relative contributions of the ground-truth and teacher probability distributions remain approximately unchanged.

Remarks. The proposed HDL method is based on existing distillation techniques that have been explored in varying context and problems [94, 96, 133, 132]. However, they are rarely jointly modelled in a unified model. Therefore, their complementary effects remain largely unknown. Moreover, the efficiency issue in person search is under-studied significantly, let alone exploiting the knowledge distillation notion. One main reason is that existing joint learning person search models [84] are dramatically inferior, therefore lacking a strong teacher model to enable the knowledge distillation. We overcome this obstacle to person search and further explore the potential of three fundamental distillation algorithms jointly for addressing the ignored and realistically significant scalability issue.

4.2.4 Model Training

As the conventional knowledge distillation, we start with training the teacher model, followed by student training using the proposed HDL algorithm.

Teacher Model. By joint learning person search, the loss function for the teacher network Θ_T is formulated as:

$$\mathcal{L}(\Theta_T) = \mathcal{L}_{ID}(\Theta_T) + \mathcal{L}_{DET}(\Theta_T), \tag{4.9}$$

where $\mathcal{L}_{ID}()$ is the cross-entropy loss for person identity classification, and $\mathcal{L}_{DET}()$ the person detection loss including box regression and binary-class classification.

Student Model. To train the student model, we also exploit the HDL loss functions in addition to the joint learning person search loss that same as Eq equation 4.9. This aims to transfer the already-trained teacher's knowledge. Formally, the loss function of the student model Θ_S is designed as:

$$\begin{aligned}\mathcal{L}(\Theta_S) =&(1-\lambda_0)*\mathcal{L}_{ID}(\Theta_S)+\lambda_0*\mathcal{L}_{PD}(\Theta_S)+\\ &\lambda_1*\mathcal{L}_{AD}(\Theta_S)+\lambda_2*\mathcal{L}_{FD}(\Theta_S)+\\ &\mathcal{L}_{DET}(\Theta_S),\end{aligned} \tag{4.10}$$

where $\lambda_{0/1/2}$ are three loss weighing hyper-parameters, estimated by cross-validation.

4.2.5 Network Architecture Details

In this section , we provide the details of HDL network architecture.

Teacher Model. We adopt a ResNet50 [3] as the stem network for the teacher model. It consists of four blocks (named conv2_x to conv5_x) each containing 3, 4, 6, 3 residual units. In particular, we choose the first layer (conv1_x, i.e. $64\times7\times7$ conv layer) to the third block (conv4_x) as feature sub-network, and conv5_x as person re-id sub-network. The person detection sub-network is built on conv4_x. The channel dimensions for the four blocks (Fig 6.4) are $c_T^0 = 256$, $c_T^1 = 512$, $c_T^2 = 1,024$, and $c_T^3 = 2,048$, respectively.

Student Model. For the student model, we use a $32\times3\times3$ conv layer with stride 2 as the input layer. To achieve a good balance between efficiency and accuracy, we construct the corresponding four blocks by setting $c_S^0 = 128$, $c_S^1 = 256$, $c_S^2 = 384$, and $c_S^3 = 512$. In each building block (Figure 4.2), we set the strides as $s_1 = s_2 = 1$ and $s_3 = 2$.

Attention Module. For both teacher and student models, we introduce a ARM unit at the end of each block (Fig 6.4). This forms an attention pyramid for richer salience learning.

4.3 Experiments

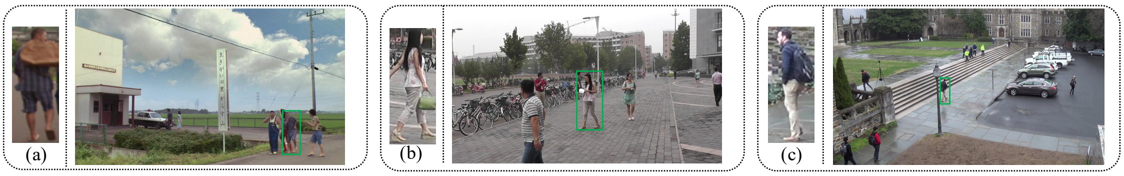


Figure 4.4: Example query and unconstrained scene images from (a) CUHK-SYSU [84], (b) PRW [85], and (c) DukeMTMC-PS [42].

Datasets. To evaluate the proposed HDL model, we used three person search benchmarks: CUHK-SYSU [84], PRW [85], and DukeMTMC-PS which is newly introduced based on the

Dataset	IDs	Images	ID Split		Image Split	
			Train	Test	Train	Test
CUHK-SYSU	8,432	18,184	5,532	2,900	11,206	6,978
PRW	932	11,816	482	450	5,704	6,112
DukeMTMC-PS	1,404	35,543	702	702	16,362	17,350

Table 4.1: Data statistics of person search datasets.

DukeMTMC tracking dataset [42]. Example images are shown in Fig 4.4. We adopted the standard evaluation setting of *CUHK-SYSU* and *PRW* (Table 4.1). We re-purposed the DukeMTMC data into a person search benchmark *DukeMTMC-PS*. The train/test ID split follows the person re-id counterpart [61]. This dataset provides much more training and test scene images than CUHK-SYSU and PRW, representing a more realistic and more challenging person search scenario. We will publicly release the DukeMTMC-PS dataset.

Performance Metrics. For person detection, a bounding box was considered as correct if the overlap with the ground truth is over 50% [84, 85]. For person re-id, we used the Cumulative Matching Characteristic (CMC) and mean Average Precision (mAP). To evaluate the model inference efficiency, we adopted the common measurement of floating point operations (FLOPs) consumed by processing one typical scene image and one person bounding box.

Competitors. For model performance comparisons, we considered six state-of-the-art deep learning person search methods, including four *joint learning* model (OIM [84], RCAA [86], IAN [87], NPSM [88]) and two *independent learning* models (MGTS [90], CLSA [89]). We did not include other significantly inferior hand-crafted feature based alternative approaches in terms of both model performance and inference efficiency.

Implementation Details. We conducted the experiments in the PyTorch framework. For model training, we adopted the SGD algorithm with the momentum set to 0.9, the weight decay to 0.0005. We set batch size to 8 for CUHK-SYSU with input size of 800×800 and 4 for PRW and DukeMTMC-PS with input size of 1920×1080. Mean value padding was used for organising images into batches. For teacher model training, we set the epoch number to 60 and initialised the learning rate at 0.005, with a decay factor of 10 at 50-th epoch. For student model training, we set the epoch number to 150 and initialised the learning rate at 0.005, with a decay factor of 5 every 50 epochs. We set the weights $\lambda_0 = 0.9$, $\lambda_1 = 2 \times 10^4$, $\lambda_2 = 2 \times 10^3$ (Eq equation 4.10), and the temperature $t = 4$ (Eq equation 4.8) by cross-validation for all the experiments. The L_2 normalisation was applied before computing the pairwise cosine similarity for re-id matching.

4.3.1 Further Analysis and Discussions

Attention Learning. We evaluated the benefits of our attention learning design. It is evident from Table 4.2 that, both the teacher and student models benefit significantly. In particular, our attention learning not only improves the quality of teacher's knowledge, but also facilitates the knowledge transfer process given that the student acquires more gains in most cases. This verifies our design consideration of integrating attention with feature and prediction in HDL.

Dataset	CUHK		PRW		Duke	
Metric (%)	Rank-1	mAP	Rank-1	mAP	Rank-1	mAP
T(w/o A)	68.5	63.8	62.1	26.3	69.9	44.3
T(w/ A)	**73.2**	**69.7**	**69.2**	**33.6**	**74.3**	**50.0**
Gain	+4.7	+5.9	+7.1	+7.3	+4.4	+5.7
S(w/o A)	42.6	38.6	50.6	16.8	56.5	26.2
S(w/ A)	**49.5**	**45.1**	**59.1**	**22.8**	**61.4**	**33.4**
Gain	+6.9	+6.5	+8.5	+6.0	+4.9	+7.2

Table 4.2: Evaluating attention (A) learning. T: Teacher; S: Student. Setting: The gallery size for CUHK-SYSU is 6,978.

Knowledge Distillation. We examined the effect of different distillation and their combinations on CUHK-SYSU. Table 4.3 reveals a couple of observations: (1) Each distillation *alone* brings about model improvements, with prediction distillation contributing the most. This is because as the model output the prediction encodes the most discriminative abstraction information. (2) As the low-level knowledge, transferring attention and feature further enhances model learning on top of high-level prediction distillation. This verifies the complementary benefits of exploiting different model knowledge in HDL design.

	Distillation			CUHK-SYSU	
	FD	AD	PD	Rank-1	mAP
1	-	-	-	49.5	45.1
2	✓	-	-	58.1	54.1
3	-	✓	-	52.0	47.8
4	-	-	✓	65.8	62.6
5	✓	✓	-	59.4	55.8
6	✓	-	✓	66.4	63.0
7	-	✓	✓	68.2	65.4
8	✓	✓	✓	**70.0**	**66.4**

Table 4.3: Evaluating different distillation. T: Teacher; S: Student. FD: Feature Distillation; AD: Attention Distillation; PD: Prediction Distillation. Setting: The gallery size for CUHK-SYSU is 6,978.

4.3.2 Comparisons to State-of-the-Art Methods

Evaluation on CUHK-SYSU. We reported the person search performance on CUHK-SYSU with the standard gallery size of 100 scene images in Table 4.5. We made the following observations: **(1)** Our teacher model HDL(T) achieves the second best rank-1 rate and mAP among all competitors. In particular, the margin of HDL(T) over all existing joint learning competitors are consistently significant. This suggests that the joint learning strategy is *not necessarily* inferior to independent learning, even without adopting sophisticated techniques like attention inference [88] and reinforcement learning [86]. **(2)** By the proposed distillation method, our student model HDL(S) can achieve very competitive performance, e.g. matching the state-of-the-art CGPS [92] and surpassing all other existing joint learning methods and one independent learning model MGTS [90]. This indicates the efficacy of the proposed distilling method in transferring the teacher's knowledge. **(3)** The proposed HDL(S) reaches the best model inference efficiency, i.e. the superior cost-effectiveness benefits over all the alternative solutions. Note, we do not evaluate the model inference cost for NPSM [88], RCAA [86] and QEEPS [91] due to their query-specific search design, a less scalable strategy than query-independent search by all the other methods. **(4)** HDL(S) is over one order of magnitude more efficient than all existing methods, which facilitates large scale and cost-effective deployments.

Type	Metric (%)	Rank-1	mAP	Cost (scene/person)
IL	MGTS [90]	83.7	83.0	>1725.6G/52.8G
	CLSA [89]	**88.5**	**87.2**	>410.7G/26.4G
JL	OIM [84]	78.7	75.5	410.7G/2.0G
	IAN(R101) [87]	80.5	77.2	1146.2G/2.0G
	NPSM [88]	81.2	77.9	-
	RCAA [86]	81.3	79.3	-
	QEEPS [91]	84.4	84.4	-
	CGPS [92]	86.5	84.1	410.7G/2.0G
	HDL(T)	87.3	86.0	427.5G/2.1G
	HDL(S)	86.2	84.6	**37.5G/76.4M**

Table 4.4: Performance evaluation on **CUHK-SYSU**. The gallery size is 100. IL: Independent Learning; JL: Joint Learning; T: Teacher; S: Student; R101: ResNet-101; G: GFLOPs (1×10^9); M: MFLOPs (1×10^6).

We further tested the model performance with the full gallery size at 6,978. This allows to evaluate larger scale search performance. Following the previous works, we compared the mAP results. Figure 4.5 shows similar observations as in Table 4.5, suggesting that the model performance advantages of HDL generalise to large scale search.

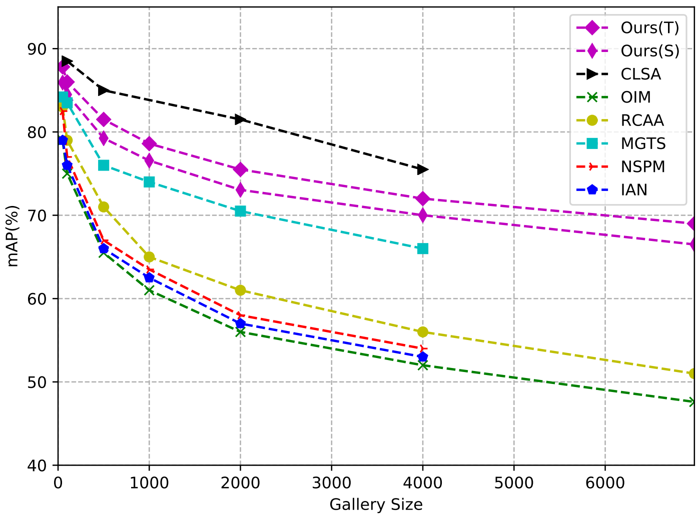


Figure 4.5: Test mAP of varying gallery sizes on CUHK-SYSU.

Evaluation on PRW. We compared the model performance on the PRW benchmark. Overall, we obtained similar comparison observations that our teacher model HDL(T) achieves the second best performance in both rank-1 and mAP rates. HDL(S) similarly approaches the accuracy levels of HDL(T) whilst significantly outperforming all existing joint learning competitors in addition to a great model efficiency advantage. This consistently indicates the cost-effectiveness and scalability superiority of our model over the existing person search methods in a more challenging application scenario.

Type	Metric (%)	Rank-1	mAP	Cost (scene/person)
IL	MGTS [90]	**72.1**	32.6	>1725.6G/52.8G
	CLSA [89]	65.0	**38.7**	>1330.7G/26.4G
JL	OIM [84]	49.9	21.3	1330.7G/2.0G
	IAN(R101) [87]	61.9	23.0	3713.7G/2.0G
	NPSM [88]	53.1	24.2	-
	HDL(T)	69.2	33.6	1381.6G/2.1G
	HDL(S)	64.4	28.2	**121.4G/76.4M**

Table 4.5: Performance evaluation on **PRW**. IL: Independent Learning; JL: Joint Learning; T: Teacher; S: Student; R101: ResNet-101; G: GFLOPs (1×10^9); M: MFLOPs (1×10^6).

Evaluation on DukeMTMC-PS. We further evaluated the performance of our HDL model on the newly introduced DukeMTMC-PS benchmark. Compared to CUHK-SYSU and PRW, test scene images from this benchmark are more than two times larger, therefore presenting a more challenging person search task. We compared with the only scalable joint learning competitor OIM and an independent learning baseline using Faster R-CNN+ResNet-50. The results in Table 4.6 show the consistent performance and efficiency superiority of HDL and the knowledge distillation efficacy from the stronger teacher model to the lightweight student model. Encouragingly, HDL(S) even surpasses the independent learning model, Faster R-CNN+ResNet50, by 2.9% (71.8-68.9) in Rank-1 and 2.8% (45.5-42.7) in mAP, in addition to more than one order of magnitude inference efficiency advantage.

Type	Metric (%)	Rank-1	mAP	Cost (scene/person)
IL	FRCNN+R50	68.9	42.7	>1330.7/26.4G
JL	OIM [84]	50.5	34.5	1330.7G/2.0G
	HDL(T)	**74.3**	**50.0**	1381.6G/2.1G
	HDL(S)	71.8	45.5	**121.4G/76.4M**

Table 4.6: Performance evaluation on **DukeMTMC-PS**. IL: Independent Learning; JL: Joint Learning; T: Teacher; S: Student; FRCNN+R50: Faster R-CNN + ResNet-50; G: GFLOPs (1×10^9); M: MFLOPs (1×10^6).

4.4 Summary

In this chapter, we present a novel *Hierarchical Distillation Learning* (HDL) method for person search in unconstrained surveillance scene images. This method is designed particularly for addressing the largely ignored *scalability* problem in person search. It is in contrast to existing alternative methods that typically focus on model performance improvement alone. Specifically, we formulate a comprehensive knowledge distillation method for transferring feature representation, attention map, and class prediction from a strong and heavy teacher model to a weak and lightweight student model. This addresses the hard-to-optimise challenge for small models. We also contribute a simple and powerful joint learning teacher model, potentially motivating the further development of new models of its kind. Extensive comparative evaluations have been conducted on three large person search benchmarks. The results validate the scalability advantages of our HDL model over a variety of state-of-the-art person search methods. We provide in-depth component analysis to give the insights on model performance gain and design rationale.

Chapter 5

Scalable Neural Architecture Search

5.1 Overview

In this chapter, we consider that the common limitation in Neural Architecture Search (NAS) is the weak capability of modelling the topological knowledge of the network architecture when constructing the continuous search space. As an intrinsic property of neural network in specific and directed acyclic graph (DAG) in general, topology plays a fundamentally crucial role in the process of NAS (Figure 5.1). To solve this limitation, we propose the notion of Neural Graph Embedding (NGE) for neural architecture search. In particular, NGE elegantly enables integrating the Graph Convolutional Network (GCN) [139] with existing solutions, including the efficient gradient-based paradigm (Figure 5.1), therefore allowing for modelling the topology of the network architecture by recursive message propagation among nodes in a cell. Importantly, through dedicated neural graph embedding we obtain a *continuous* search space in a principled manner. This not only facilitates the NAS design, but also enjoys favourable search efficiency even compared with existing fast GD-based NAS methods like DARTS [14] and NAO [24].

5.2 NAS as Optimisation

Following DARTS [14], the search problem can be *efficiently* formulated in a gradient differentiable manner by relaxing the search space to be continuous.

Continuous Relaxation. For a connection from the i-th node to the k-th node in a cell with the *architecture parameters* $\boldsymbol{a}^{(i,k)}$, a softmax over all possible operations is applied to obtain the

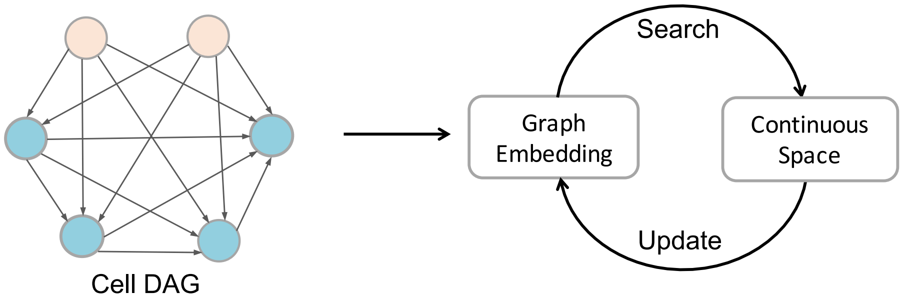


Figure 5.1: The concept of neural graph architecture search. We represent the cell of a network architecture with *directed acyclic graph* (DAG), which enables the search space to be represented in a continuous space, and facilitates the adoption of gradient descent based optimisation.

categorical choice of a particular operation:

$$\overline{o}^{(i,k)}(F_i) = \sum_{o \in \mathbb{O}} \frac{\exp\left(a_o^{(i,k)}\right)}{\sum_{o' \in \mathbb{O}} \exp\left(a_{o'}^{(i,k)}\right)} o(F_i). \tag{5.1}$$

Optimisation. Within a continuous search space, a common search process for neural architecture is generally composed of two separate optimisation procedures. Given the network parameter space $\mathbb{W}$ and the architecture space $\mathbb{A}$, the first procedure (Eq. (6.11)) discovers the optimal parameters $\boldsymbol{w}^* \in \mathbb{W}$ for a given architecture $\boldsymbol{a} \in \mathbb{A}$ w.r.t a training objective function $\mathcal{L}_{train}$:

$$\boldsymbol{w}^*(\boldsymbol{a}) = \arg\min_{\boldsymbol{w}} \mathcal{L}_{train}(\boldsymbol{w}, \boldsymbol{a}). \tag{5.2}$$

The second procedure (Eq. (6.12)) then explores the optimal architecture $\boldsymbol{a}^*$ over the architecture space $\mathbb{A}$ w.r.t a validation objective function $\mathcal{L}_{val}$:

$$\boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} \mathcal{L}_{val}(\boldsymbol{w}^*(\boldsymbol{a}), \boldsymbol{a}). \tag{5.3}$$

Once this alternated optimisation is done, an amenable cell architecture is deviated by retaining the top-k strongest incoming operations from all the previous nodes.

5.3 Methodology

We aim to make full use of the intrinsic topology information of neural networks for facilitating the optimisation of NAS. To this end, we propose the notion of *Neural Graph Embedding* (NGE). The idea is that, we represent the cell and block structures with a neural graph, and leverage Graph Convolutional Networks (GCN) [139] to form the relational embeddings of this neural graph. Not only does our method capture the underlying topology information of network architecture comprehensively, but it also creates a means of representing the discrete operator selection by continuous feature vectors that substantially facilitate the optimisation of operator association. An overview of the proposed NGE model is depicted in Fig. 6.4.

In the followings, we first describe how to build a search space as a graph. We then provide the detail of GCN in the context of neural architecture graph. Finally, we delve into the details of how we integrate NGE into the task of NAS.

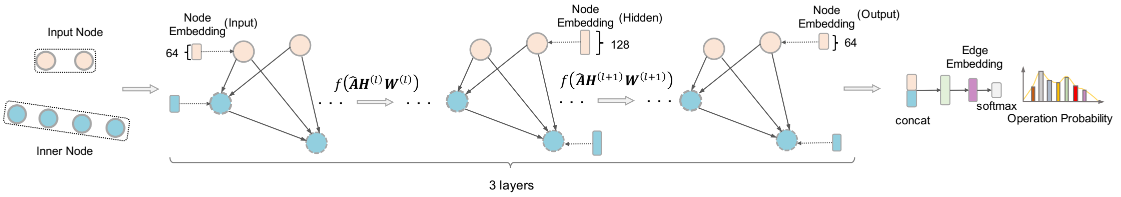


Figure 5.2: An overview of the proposed Neural Graph Embedding (NGE) for NAS. Each *node* denotes a computational transformation in a cell, initialised as a one-hot vector sequentially. We use a 3-layers GCN to perform the propagation of node-to-node interaction information. Each *edge* represents a connection between two nodes. We represent it by mapping the concatenation of the embeddings of the two nodes with a multilayer perceptron (MLP). It is this edge representation that significantly facilitates the optimisation of operation selection, e.g. learning an operation class classifier end-to-end.

5.3.1 A Neural Graph

Rather than searching over a search space $\mathbb{A}$ *directly*, we transform the architecture search space in the form of a graph. This forms a *neural graph*, leading to two advantages: (1) It explicitly encodes the high-order relationships between different nodes in a cell; (2) It also implicitly regulates the relationships between nodes and operations.

Search Space as Neural Graph. As discussed above, given the factorised search space, all we need for NAS is to search an appropriate design of blocks in a cell. Intrinsically, a cell with N ordered nodes can be defined as a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v \in \mathcal{V}$ has an associated embedding vector $\boldsymbol{x}_v \in \mathcal{X}$ (Note that the output node is excluded for consideration as its incoming connections are fixed); And the edge $e_{u \to v} = (u, v) \in \mathcal{E}$ is the connection between node u and node v, representing the information flow $u \to v$. Besides, a specific operation $o_{u,v}$ from the candidate set $\mathbb{O}$ is applied to the edge $e_{u \to v}$. Forming this *neural graph* search space $\mathcal{G}$, next we aim to learn continuous embeddings (representations) for the nodes and the edges of $\mathcal{G}$.

Node Embedding. We learn the node embedding by designing a Graph Convolutional Network (GCN). This allows naturally modelling the topological relationships between nodes. Specifically, the input to each node v is an initial embedding vector $\boldsymbol{x}_v$, initialised as a specific one-hot vector different for each node and updated simultaneously during learning. We summarise the inputs of all the nodes as a matrix $\boldsymbol{X} = \left[\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{|\mathcal{V}|}\right] \in \mathbb{R}^{|\mathcal{V}| \times D}$, where D denotes the dimension of the input embedding. The GCN outputs a node-level representation $\boldsymbol{Z} = \left[\boldsymbol{z}_1, \cdots, \boldsymbol{z}_{|\mathcal{V}|}\right] \in \mathbb{R}^{|\mathcal{V}| \times F}$, where F denotes the dimension of the output embedding.

Formally, the node embedding learning is formulated as:

$$\boldsymbol{Z} = \texttt{GCN}(\boldsymbol{X}; \Theta_n), \tag{5.4}$$

where Θ_n is the parameter for the GCN model. More specifically, considering $|\mathcal{V}|$ ordered nodes, the structure of graph search space is represented as a normalised adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ as the follows:

$$A_{i,j} = \begin{cases} \frac{1}{i+1} & \text{if } i < j \,\&\, i > 1, \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

To incorporate self-reinforcement, we further form an augmented version $\hat{\boldsymbol{A}}$ by:

$$\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}, \tag{5.6}$$

where $\boldsymbol{I} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is the identity matrix. Let $f(\cdot)$ denotes the ReLU activation function, we then formulate the per-layer learning module as:

$$\boldsymbol{H}^{(l+1)} = f(\hat{\boldsymbol{A}} \boldsymbol{H}^{(l)} \boldsymbol{W}^{(l)}), \tag{5.7}$$

where $\boldsymbol{W}^{(l)} \in \Theta_n$ is the l-th layer's parameter, and L the total layer number. In particular, $\boldsymbol{H}^{(0)} = \boldsymbol{X}$ and $\boldsymbol{H}^{(L)} = \boldsymbol{Z}$. With this formulation, the topological knowledge between different nodes can be continuously propagated in a stack of feed-forward operations sequentially, enabling to reveal high-order relationships across the whole neural graph.

Edge Embedding. We learn the edge embedding based on the embeddings of the two associated nodes. Specifically, for an edge between the i-th and j-th nodes, we first concatenate their embeddings $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ to merge their information. Then, we deploy an efficient MLP with

```
Algorithm 1: Neural Graph Embedding (NGE) for NAS
Input: Training set: $\mathcal{X}_{train}$; validation set: $\mathcal{X}_{val}$.
Output: Network architecture $\boldsymbol{a}^*$
$\boldsymbol{w}$, $\boldsymbol{X}$, $\Theta_n$, $\Theta_e$, $\Theta_o$ ← random initialisation
for Num. of max epochs. do
    $\boldsymbol{Z}$ ← obtain by Eq. (5.4);
    $\boldsymbol{E}$ ← obtain by Eq. (5.8);
    $\boldsymbol{P}$ ← obtain by Eq. (5.9);
    for samples in $\mathcal{X}_{train}$ do
        Update weights $\boldsymbol{w}$ by descending $\nabla_{\boldsymbol{w}}\mathcal{L}_{train}(\boldsymbol{w},\boldsymbol{P})$;
    for samples in $\mathcal{X}_{val}$ do
        $\boldsymbol{Z}$ ← obtain by Eq. (5.4);
        $\boldsymbol{E}$ ← obtain by Eq. (5.8);
        $\boldsymbol{P}$ ← obtain by Eq. (5.9);
        Update $\boldsymbol{X}$, $\Theta_n$, $\Theta_e$, $\Theta_o$ by descending $\nabla\mathcal{L}_{val}(\boldsymbol{w},\boldsymbol{P})$;
Derive the final architecture $\boldsymbol{a}^*$ based on the learned $\boldsymbol{P}$.
```

two fully-connected (FC) layers and ReLU activation to further learn the edge embedding $\boldsymbol{e}^{(i,j)}$. Formally, we formulate the edge embedding learning as the follows:

$$\boldsymbol{e}^{(i,j)} = \texttt{MLP}(\texttt{concat}(\boldsymbol{z}_i, \boldsymbol{z}_j); \Theta_e) \in \mathbb{R}^K, \tag{5.8}$$

where Θ_e is the parameter set of the edge embedding MLP, shared for all the edges, and K denotes the dimension of edge embedding. Importantly, the edge embedding $\boldsymbol{e}^{(i,j)}$ not only encodes the *local* pairwise relationships between the i-th and j-th nodes, but also considers the *global* higher-order relationships among all the nodes. In doing so, we provide a principled method for modelling the comprehensive topological knowledge of a neural architecture.

5.3.2 NGE for NAS

Applying the NGE to the NAS task is straightforward, since an edge $e_{u\to v} = (u, v) \in \mathcal{E}$ can be readily associated with the operation selection. That being said, this allows us to derive the optimal operation selection from the edge embeddings $\boldsymbol{E}$ in a standard learning framework. It is worth mentioning that our NGE is a *general representation model* that can be integrated into different NAS paradigms. For RL-based NAS, we can compute the actions from the edge embeddings for choosing operations. For EA-based NAS, the edge embeddings act as a controller to generate mutations. Due to the resource constraint we focus on the efficient GD-based NAS. Concretely, we predict the operation probability distribution for all the relaxed connections by using the NGE edge embeddings as input.

Operation Probability. Given the edge embedding $\boldsymbol{e}^{(i,j)}$ between the i-th and j-th nodes, we compute the associated operation probability $\boldsymbol{p}^{(i,j)} \in \mathbb{R}^{O}$ by a FC layer with the softmax activation:

$$\boldsymbol{p}^{(i,j)} = \texttt{softmax}(\texttt{FC}(\boldsymbol{e}^{(i,j)};\Theta_o)), \tag{5.9}$$

where Θ_o is the parameter for the FC layer and shared for all the edges. We summarise the operation probability of all the edges as a matrix $\boldsymbol{P} = \left[\boldsymbol{p}_1, \cdots, \boldsymbol{p}_{|\mathcal{E}|}\right] \in \mathbb{R}^{|\mathcal{E}| \times O}$. We reformulate the continuous relaxation in Eq. (5.1) as:

$$\bar{o}^{(i,k)}(\mathsf{F}_i) = \sum_{o \in \mathbb{O}} p_o^{(i,k)} o(\mathsf{F}_i). \tag{5.10}$$

In this way, we can integrate the NGE learning into an existing GD-based NAS framework seamlessly.

Learning. In the search process, we jointly learn the NGE and the network parameters $\boldsymbol{w}$ in a fully differentiable manner. Unlike DARTS [14] optimising for each batch input, we formulate the optimisation in an epoch-wise way, which would provide better converge speed (see a comparison in experiments). The pseudo code of NES for NAS is summarised in Algorithm 1.

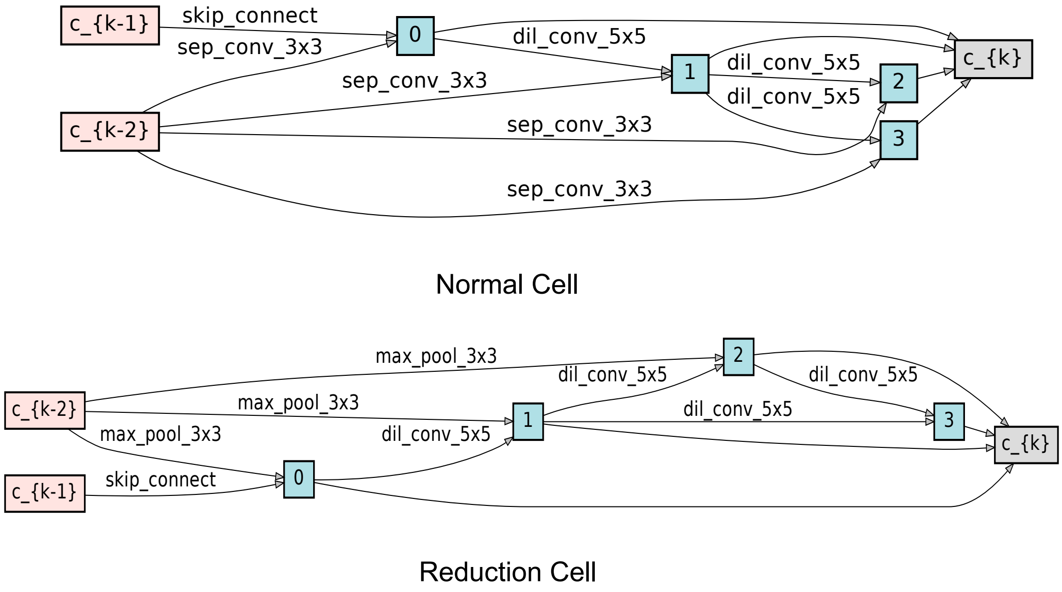
5.4 Experiments

To show the effectiveness and transferability of our NGE method on both image classification and semantic segmentation tasks, we conducted the network architecture search on CIFAR-10 *only*, and compared the obtained architecture with both state-of-the-art human-design and NAS models on CIFAR-10, CIFAR-100, ImageNet and PASCAL VOC 2012 datasets. Below we gave the experiment details including datasets, model instantiating, evaluation, and analysis.

5.4.1 Datasets

CIFAR. Both CIFAR-10 and CIFAR-100 [140] contain 50K/10K train/test RGB images with a unified resolution of 32×32. The images of both datasets are categorised into 10 and 100 classes, respectively.

ImageNet. For the large-scale image classification evaluation, we used the ILSVRC2012, a subset of ImageNet [2] that contains 1K classes, 1.28M training images, and 50K validation



Normal Cell

Reduction Cell

Figure 5.3: Normal cell and reduction cell obtained by NGE.

samples.

PASCAL VOC 2012. We used the PASCAL VOC 2012 [141] for semantic segmentation evaluation. It consists 1,464/1,449/1,456 train/val/test images with pixel annotation from 21 classes. Extra annotations from [142] were used for data augmentation, resulting in 10,582 training images. We used mean pixel intersection-over-union (mIOU) across all the classes to measure the performance.

5.4.2 NGE Instantiating

We constructed the graph search space $\mathcal{G}=(\mathcal{V},\mathcal{E})$ with $|\mathcal{V}|=6$ nodes (2 input nodes and 4 inner nodes). As such, there are $|\mathcal{E}|=14$ edges totally. For the node embedding, we set the input dimension $D=64$ and the output dimension $F=64$. We used a ($L=3$)-layers GCN with the hidden dimensions of 128. A MLP with 2-FC layers at the dimension of 64 was applied to learn the edge embedding with dimension $K=64$, taking as input the concatenation of two node embeddings. We included $O=7$ primitive operations in the candidate function set $\mathbb{O}$ as introduced early. All the parameters (Θ_n, Θ_e and Θ_o) were randomly initialised in the normal distribution. All the FC layers use no bias.

5.4.3 Cell Search

We followed the setup of existing methods [13, 14, 15] to search the convolutional cells on CIFAR-10. A small proxy network consists of 8 cells was constructed for searching both the normal cell and the reduction cell. As shown in Fig. 2.2(b), two reduction cells are located at the $1/3$ and $2/3$ of the total depth of the network. The detailed head structure for CIFAR is depicted in Fig. 2.2(c), in which the number of initial channels is 16. We split 25K images from the training set for validation. We initialised the node embeddings $\boldsymbol{X}_{normal}$ and $\boldsymbol{X}_{reduction}$ for the normal cells and reduction cells, where $\boldsymbol{X}_{normal}$ is shared for all normal cells and $\boldsymbol{X}_{reduction}$ was shared for all reduction cells. For the network parameter $\boldsymbol{w}$, we used SGD with an initial learning rate 0.025 and the momentum of 0.9. We decayed the learning rate to 0 during training using a cosine schedule. A weight decay of 3×10^{-4} was imposed to avoid over-fitting. For the NGE learning, we used the Adam optimiser with a fixed learning rate 6×10^{-4} and set the weight decay to 1×10^{-3}. To search the normal cell and reduction cell efficiently, we used 25 epochs for training the proxy network. With NGE, the search on CIFAR-10 took only 2.4 hours on a single NVIDIA Tesla V100 GPU. The searched cells by NGE is shown in Fig. 5.3.

5.4.4 Further Analysis

To further demonstrate the necessity of learning neural graph embeddings for NAS, we compared three alternative learning strategies: **(1)** As the *baseline* method, without neural embeddings we directly learn the architecture parameters as DARTS [14]. **(2)** A *plain* embedding learning strategy is added on the baseline, in which no relationships between nodes is modelled. **(3)** Two *RNN* layers are further introduced upon the plain method to model the relationships between nodes in a sequential manner. For fair comparison, we followed the same setting as NGE to search the cell architectures on CIFAR-10 and reported the final performance on both CIFAR-10 and CIFAR-100. Table 5.3 shows that: (1) Compared with the baseline, learning embedding for NAS consistently helps find better cell architectures. (2) Modelling the relationships between nodes can further benefit the searching of cell architectures. (3) The proposed NGE outperforms alternative learning strategies significantly. This verifies the significance of graph embeddings for NAS and the superiority of our method.

Architecture	Venue	Error (%)		Params	Search Cost		Search Method
		C10	C100	(M)	GPUs	Days	
DenseNet-BC [44]	CVPR17	3.46	17.18	25.6	-	-	manual
NASNet-A + cutout [10]	CVPR18	2.65	-	3.3	450	1800	RL
AmoebaNet-A + cutout [13]	CVPR18	3.34	-	3.2	450	3150	EA
AmoebaNet-B + cutout [13]	CVPR18	2.55	-	2.8	450	3150	EA
Hireachical Evolution [12]	ICLR18	3.75	-	15.7	200	300	EA
PNAS [15]	ECCV18	3.41	-	3.2	100	1.5	SMBO
ENAS + cutout [11]	ICML18	2.89	-	4.6	1	0.5	RL
ProxylessNAS-G + cutout [18]	ICLR19	**2.08**	-	5.7	-	4	GD
RENAS [143]	CVPR19	2.88	-	3.5	4	1.5	EA&RL
DARTS (1st) + cutout [14]	ICLR19	3.00	-	3.3	1	1.5	GD
DARTS (2nd) + cutout [14]	ICLR19	2.76	17.54	3.3	1	4.0	GD
SNAS (mild) + cutout [23]	ICLR19	2.98	-	2.9	1	1.5	GD
SNAS (moderate) + cutout [23]	ICLR19	2.85	-	2.8	1	1.5	GD
SNAS (aggressive) + cutout [23]	ICLR19	3.10	-	**2.3**	1	1.5	GD
GHN + cutout [144]	ICLR19	2.84	-	5.7	1	0.84	GD
GDAS [C=36,N=6] [145]	CVPR19	2.93	18.38	3.4	1	0.84	GD
GDAS(FRC) [C=36,N=6] [145]	CVPR19	2.82	18.13	2.5	1	0.68	GD
BayesNAS(0.010) + cutout [146]	ICML19	3.02	-	2.5	1	0.2	GD
BayesNAS(0.007) + cutout [146]	ICML19	2.90	-	3.1	1	0.2	GD
BayesNAS(0.005) + cutout [146]	ICML19	2.81	-	3.4	1	0.2	GD
ASNG-NAS + cutout [147]	ICML19	2.83	-	3.9	1	0.11	GD
NGE + cutout	Ours	2.60	**16.53**	3.5	1	**0.1**	GD

Table 5.1: Comparisons with the state-of-the-art architectures on the CIFAR-10 and CIFAR-100 datasets.

5.4.5 Architecture Evaluation

CIFAR. To measure the final image classification performance of the searched cells on CIFAR-10 and CIFAR-100, an evaluation network of 20 cells, 36 initial channels and an auxiliary tower with loss weight 0.4 was created. The network was trained from scratch for 600 epochs with 128-sized mini-batches. To avoid over-fitting, the cutout regularisation [150] with length 16 and the drop-path [151] of probability 0.3 were applied. The weight decay values for CIFAR-10 and CIFAR-100 were set to 3×10^{-4} and 5×10^{-4} individually. For model training, the standard SGD optimisation with a momentum of 0.9 was performed. The initial learning rate was 0.25, decayed to 0 with a cosine scheduler.

We summarised the evaluation results with comparison to the state-of-the-art methods in Table 6.2. Using NGE, the discovered network with 3.5M parameters achieves 2.60% error rate on CIFAR-10. Without re-searching, we applied the same network on CIFAR-100 and achieved 16.53% error rate. We made three observations: **(1)** NGE achieves a very competitive result

Architecture	**Venue**	**Test Err. (%)**		**Params**	**×+**	**Search Cost**	**Search Method**
		top-1	**top-5**	**(M)**	**(M)**	**(GPU-days)**	
MobileNet-v1(1.0)[62]	arXiv17	29.4	10.5	4.2	575	-	manual
MobileNet-v2(1.0)[136]	CVPR18	28.0	-	3.4	300	-	manual
ShuffleNet 2× (v1) [148]	CVPR18	26.4	10.2	≈5	524	-	manual
ShuffleNet 2× (v2) [149]	ECCV18	25.1	-	≈5	591	-	manual
NASNet-A [10]	CVPR18	26.0	8.4	5.3	564	1800	RL
NASNet-B [10]	CVPR18	27.2	8.7	5.3	488	1800	RL
NASNet-C [10]	CVPR18	27.5	9.0	4.9	558	1800	RL
PNAS [15]	ECCV18	25.8	8.1	5.1	588	1.5	SMBO
AmoebaNet-A [13]	AAAI19	25.5	8.0	5.1	555	3150	EA
AmoebaNet-B [13]	AAAI19	26.0	8.5	5.3	555	3150	EA
AmoebaNet-C [13]	AAAI19	24.3	7.6	6.4	570	3150	EA
ProxylessNAS (GPU) [18]	ICLR19	24.9	7.5	7.1	**465**	8.3	GD
RENAS [143]	CVPR19	**24.3**	**7.4**	5.4	580	6	EA&RL
DARTS [14]	ICLR19	26.7	8.7	4.7	574	4.0	GD
SNAS [23]	ICLR19	27.3	9.2	4.3	522	1.5	GD
GHN [144]	ICLR19	27.0	8.7	6.1	569	0.84	GD
GDAS [C=50,N=4] [145]	CVPR19	26.0	8.5	5.3	581	0.84	GD
GDAS-F [C=52,N=4] [145]	CVPR19	27.5	9.1	4.4	497	0.68	GD
BayesNAS (0.010) [146]	ICML19	28.1	9.4	4.0	-	0.2	GD
BayesNAS (0.007) [146]	ICML19	27.3	8.4	3.3	-	0.2	GD
BayesNAS (0.005) [146]	ICML19	26.5	8.9	3.9	-	0.2	GD
NGE (searched on CIFAR10)	Ours	25.3	7.9	5.0	563	**0.1**	GD

Table 5.2: Comparisons with the state-of-the-art architectures on the ImageNet benchmark with the mobile setting.

Model	**Test Error (%)**	
	CIFAR-10	**CIFAR-100**
Baseline	3.44	17.90
Plain	3.14	17.60
RNN	2.71	16.97
NGE	**2.60**	**16.53**

Table 5.3: Comparing different embedding learning models.

(third best) on CIFAR-10, whilst enjoying the fastest search speed (only 0.1 GPU day). This demonstrates the cost-effective advantages of our NGE model, compared with ProxylessNAS [18] with the best accuracy and 4 GPU days and AmoebaNet-B [13] with the second best accuracy and 3150 GPU days. **(2)** Compared with GHN which also conducts graph-based search, our NGE can achieve the cells with less parameters (3.4M vs 5.7M) at a significant less cost (0.1 vs 0.84 GPU day), while obtaining a better performance (2.60 vs 2.84). **(3)** Directly transferring the

CIFAR-10 searched network to CIFAR-100 can achieve the best result, outperforming DARTS [14] and GDAS [145] significantly. This indicates the superior transferability of the network searched by our method in a challenging cross-dataset evaluation.

ImageNet. To evaluate the transferability of the architectures discovered by NGE on the large scale ImageNet benchmark, we followed the mobile setting as in [14, 145], where the number of multiply-add operations is restricted to be less than 600M with the input size at 224 × 224. Specifically, we constructed an evaluation network with 14 cells and 48 initial channels; The detailed head structure consists of three conv layers, as shown in Fig. 2.2(c). An auxiliary tower with loss weight 0.4 was also applied. We trained this model using the SGD for 250 epochs at batch-size 512 on 4 Nvidia Tesla P100 GPUs. We initialised a learning rate of 0.25 and reduced it to 0 by a linear scheduler. Learning rate warmup [152] was applied for the first 5 epochs to deal with the large batch-size and learning rate.

The ImageNet results in the mobile setting are presented in Table 6.4. Notably, the cell architectures found on CIFAR-10 by our method can achieve highly competitive performance, with significantly less computation cost (0.1 GPU day *vs* 6 GPU days for RENAS and 3,150 GPU days for AmoeBaNet). Unlike ProxylessNAS searching the network on ImageNet directly using 8.3 GPU days, the network searched by NGE on CIFAR-10 can be successfully transferred. Moreover, NGE discovers the cells on CIFAR-10 that performs better on ImageNet than state-of-the-art GD-based methods (GHN, DARTS, SNAS, GDAS and BayesNAS).

Pascal VOC 2012. We further conducted a semantic segmentation experiment with DeepLabv3 [153]. In this test, the Atrous Spatial Pyramid Pooling (ASPP) module, that contains three 3 × 3 convolutions with different atrous rates, was applied. To make a fair comparison, we followed the setting as in RENAS [143] and trained on the PASCAL VOC dataset using the above ImageNet pretrained network as the backbone model. We set the output stride to 16, which is the ratio of the input to the output spatial resolution. Note, we did *not* apply multi-scale inference and left-right flipping to improve the performance.

In Table 5.4, we summarise the validation set results in two pretraining settings (ImageNet and COCO [154]) and presented comparisons with other mobile networks. The results show that: **(1)** In both settings, our model achieves the best performance with 75.96% mIOU. Unlike the two state-of-the-art manually-designed models (MobileNet-v1 and MobileNet-v2), NGE does *not* rely on the stronger COCO pretraining. **(2)** Our model outperforms other two state-of-the-art

Model	**Dataset**	**#Params**	**mIOU(%)**
MobileNet-v1 [62]	COCO	11.15M	75.29
MobileNet-v2 [136]	COCO	4.52M	75.70
MobileNet-v1 [62]	ImageNet	11.15M	68.79
MobileNet-v2 [136]	ImageNet	4.52M	70.02
NASNet-A [10]	ImageNet	12.39M	73.68
RENAS [143]	ImageNet	11.63M	75.83
NGE	ImageNet	10.31M	**75.96**

Table 5.4: Semantic segmentation evaluation with DeepLabv3 on the PASCAL VOC 2012 validation set.

NAS-designed models, whilst having less parameters (10.31M with 75.96% mIOU *vs* 12.39M with 73.68% mIOU for NASNet-A and 11.63M with 75.83% mIOU for RENAS). Overall, the cells discovered on CIFAR-10 by NGE surpass both state-of-the-art hand-crafted and NAS-mined designs on the semantic segmentation task.

5.5 Summary

We presented a generic Neural Graph Embedding (NGE) method for neural architecture search (NAS). Unlike existing methods, NGE uniquely takes into account the intrinsic topology knowledge of neural networks from the directed acyclic graph perspective. It gives rise to a generic neural network representation with the flexibility of benefiting various NAS paradigms. As an efficient showcase, we demonstrate the advantages of NGE in a gradient descent based NAS framework. Extensive experiments on image classification and semantic segmentation show that with our method, high-quality cell architectures can be identified at a significant low computation cost.

Chapter 6

Scalable Neural Operator Search

6.1 Overview

In this chapter, we aim to address the limitation of operations by extending the search space of NAS with feature self-calibration operations for scaling up the search boundary. This makes a *heterogeneous search space*. Consequently, the way of feature tensor interaction and combination is dramatically diversified, from the conventional addition operator $\oplus$ only to the combination of addition $\oplus$, multiplication $\odot$ for attention modelling, and dynamic convolution $\circledast$. In this regard,we call the proposed method ***Neural Operator Search*** (NOS).

Such a search space enhancement is critical since NAS is enabled to explore stronger and previously undiscovered network architectures, which opens a door to potentially take the NAS research to the next level. In the *no free lunch* saying, this also comes with two new challenges: (i) It is non-trivial and more challenging to assemble such heterogeneous tensors and operations (i.e. features, attentions and dynamic weights) in a unified computing block, as compared to the conventional homogeneous feature-tensor-to-feature-tensor transformation; (ii) The search space increases exponentially which leads to a much harder NAS problem. To address the first challenge, we formulate a *heterogeneous operator cell* characterised by a novel heterogeneous residual block. This block, formulated in a residual learning spirit [3], is designed specially for fusing all the different types of tensors and operations synergistically. To solve the second challenge, we propose leveraging the *attention transfer* [96] idea to facilitate the search behaviour across this significantly larger network space via following the attention guidance of a pretrained teacher model. As we will show, this guidance not only makes the search more efficient but also improves the search result.

6.2 Method

In this section, we start by formulating a *heterogeneous search space* for NAS (Section 6.2.1), followed by a dedicated *heterogeneous operator cell* to enable composing the heterogeneous operations in a unified computing block with synergistic interaction and cooperation (Section 6.2.2). To overcome the intrinsic architecture discovery challenges from more expressive search space, we further develop an *attention guided search* scheme (Section 6.2.3).

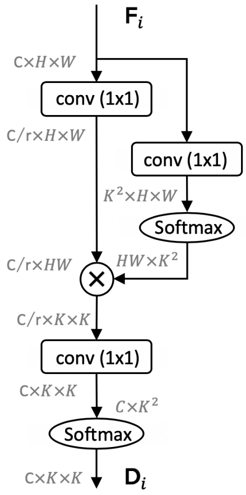


Figure 6.1: Structure of the proposed dynamic convolutions for image classification. $\otimes$ denotes matrix multiplication.

6.2.1 Heterogeneous Search Space

To enrich the NAS search space so that more advanced network architectures can be discovered, we introduce a *heterogeneous search space* $\mathbb{A}$ that considers three different types of representation learning capabilities: (1) *Feature transformations*; (2) *Attention learning*; and (3) *Dynamic convolutions*. More concretely, we form three sets of primitive computing operations that produce *features*, *attentions* and *dynamic weights*, respectively. This novel search space generalises the conventional counterpart which is limited to the first type of operations [14, 11], and incorporates the self-calibration learning capabilities (i.e. the second and third types) in NAS. Importantly, while the search space changes, the generic search strategies still apply therefore being largely open for collaborating with existing NAS methods. For instance, in the proxy-based NAS strategy we may first search for a computing cell with heterogeneous operations as the building block and then form the final network architecture by sequentially stacking multiple such cells layer-by-layer.

Next, let us describe the heterogeneous primitive operation set $\mathcal{O}$ which consists of the following three disjoint subsets: $\mathcal{O}_f$, $\mathcal{O}_a$ and $\mathcal{O}_d$, along with their aggregation or application operators.

Feature Transformation Operations $\mathcal{O}_f$. We adopt the feature transformation/learning operation set $\mathcal{O}_f$ same as in [14, 15], including the following 7 operations: 3×3 and 5×5 separable convolutions, 3×3 and 5×5 dilated separable convolutions, 3×3 average pooling, 3×3 max pooling, and identity. Every operation $o_f \in \mathcal{O}_f$ takes as input a feature tensor and outputs another feature tensor, i.e. *homogeneous* feature-tensor-to-feature-tensor transformation. For multiple feature tensor aggregation, the element-wise addition operator $\oplus$ is typically used.

Attention Learning Operations $\mathcal{O}_a$. Inspired by recent designs of attention learning modules [26, 138, 137], we form the $\mathcal{O}_a$ by considering two types of attention learning prototypes: *spatial-wise* and *channel-wise* attentions. Specifically, a *spatial-wise* attention operation learns a saliency map for an input feature tensor in order to calibrate the importance of different spatial positions. In contrast, a *channel-wise* attention operation produces a vector of scaling factors from the aggregated global context of an input tensor for adaptively calibrating the channel dependency. To enforce attentive calibration on feature tensor, the element-wise multiplication operator $\odot$ is a typical choice for both *spatial-wise* and *channel-wise* attentions.

Dynamic Convolution Operations $\mathcal{O}_d$. Dynamic convolutions, designed for the sake of self-adaptation, *generate* dynamic kernel weights in accordance with the input feature tensor. It is often in form of depth-wise separable convolution as the feature transformation operation. Tailored for either NLP or dense prediction tasks, existing dynamic convolution designs [30, 29] are not suitable for image classification with different problem nature. It hence needs to be reformulated in order to be effective for learning discriminative image representations. We consider two design principles: (i) structurally lightweight whilst (ii) functionally strong and powerful with great modelling capability.

To that end, we present a dynamic convolution structure specialised for cost-effective image classification, as shown in Figure 6.1. Concretely, it consists of three compact modules com-

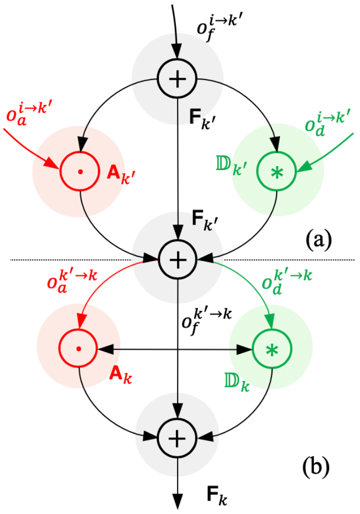


Figure 6.2: Heterogeneous Residual block for formulating the inner node computation. (a) First-tier *individual* computation; (b) Second-tier *collective* computation.

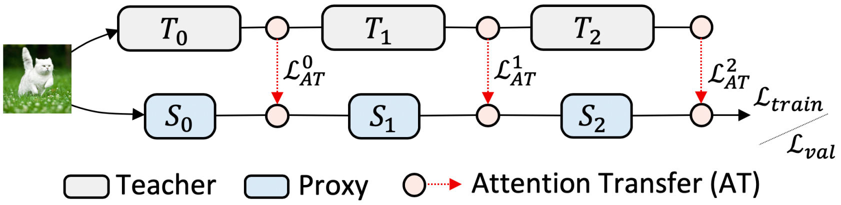


Figure 6.3: Overview of attention guided search. T_i and S_i ($i \in \{0,1,2\}$) denote the i-th stage of the teacher and proxy (student) networks.

posed in a cooperation: (a) a *bottleneck* module, to compress an input feature tensor by a ratio of r; (b) a *kernel transform* module, to learn latent representations with a kernel dimension of $k \times k$; (c) a *kernel decode* module, to read out the dynamic kernel weights with the channel dimension same as the input feature tensor. This design is motivated, in part, by the long-range dependency modeling [155, 156] and global context aggregation [26, 157], elegantly integrating their merits via a unified formulation. For the output of dynamic convolutions, we consider two common kernel sizes: 3×3 and 5×5. In a depth-wise manner, we apply a standard or dilated convolution operator $\circledast$ to transform the input feature tensor. It is noteworthy to point out that, this type of convolutional kernel is *specific* for each feature tensor of a particular image sample (i.e. dynamic), rather than learned from a training dataset and *fixed* for all the input samples (i.e. static) as the conventional convolutional operations in the feature transformation set.

6.2.2 Heterogeneous Operator Cell

Due to different natures of heterogeneous computing capabilities, a *unification* structure is needed for composing the primitive operations $\mathcal{O} = \mathcal{O}_f \cup \mathcal{O}_a \cup \mathcal{O}_d$ and aggregation/application operators $\mathcal{C} = \{\oplus,\ \odot,\ \circledast\}$ in such a way that their representation learning potentials can be well mined. To that end, we formulate a *heterogeneous operator cell*, a directed acyclic graph (DAG) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, joining conventional feature transformations and proposed self-calibration operations synergistically.

Formally, a heterogeneous operator cell consists of N ordered feature (tensor) nodes $\mathcal{V} = \{\mathsf{F}_k|, 1 <= k <= N\}$. Following [10], F_1 and F_2 are the outputs from the previous cells regarded as two *input nodes*, $\{\mathsf{F}_k\}_{k=3}^{N-1}$ denotes the *inner nodes* that perform computation, and the N-th node F_N is the cell *output node* formed as the concatenation of all the inner nodes, i.e. $\mathsf{F}_N = \texttt{concat}(\{\mathsf{F}_k\}_{k=3}^{N-1})$. The *edge* $e^{i \to k} = (i,k) \in \mathcal{E}$ specifies the connection between the i-th and k-th nodes (the information flow $i \to k$), associated with a specific operation $o^{i \to k}$ selected from the heterogeneous primitive operation set $\mathcal{O}$. The key is to design a computing block for the inner nodes with heterogeneous computations.

Heterogeneous Residual Block. It is non-trivial to design a heterogeneous computing block due to being *not* straightforward feature-tensor-to-feature-tensor transformation as in the conventional homogeneous operation. It involves self-calibrating the input feature tensor *itself* in addition to the homogeneous feature transformation. To facilitate adding the extra capacity, we formulate a heterogeneous residual block (see Figure 6.2) characterised by a *surrogate node* k' in the computing block associated with each inner node k, for enabling richer feature tensor manipulations. This is in a residual learning spirit [3], allowing to conduct self-calibration reliably.

Moreover, we design a two-tier computing hierarchy: the first tier for *individual* computation per input feature tensor to capture the specificity, and the second tier for *collective* computation on the set of all the input feature tensors as a whole to capture the intrinsic structural relations between feature tensors and the global input properties. The two tiers are connected by the surrogate node k'.

Formally, we take as input all the previous nodes $\{\mathsf{F}_i|, i < k\}$, process them separately with heterogeneous operations, and combine the processed results by summation (Figure 6.2 (a)) as:

$$F_{k'} = \sum_{i<k} o_f^{i \to k'}(F_i), \tag{6.1}$$

$$A_{k'} = \sum_{i<k} o_a^{i \to k'}(F_i), \tag{6.2}$$

$$\mathbb{D}_{k'} = \left\{ o_d^{i \to k'}(F_i) \right\}_{i<k} \tag{6.3}$$

where $F_{k'}$, $A_{k'}$, and $\mathbb{D}_{k'}$ are the three types of intermediate outputted tensors, i.e. features, attentions, and dynamic weights, respectively. These are subsequently aggregated into an *intermediate calibrated tensor*, i.e. the surrogate node $F_{k'}$, using element-wise addition in-between on feature self-calibration and transformation as:

$$F_{k'} = \underbrace{F_{k'}}_{feature} \oplus \underbrace{(F_{k'} \odot A_{k'})}_{attention} \oplus \underbrace{\sum_{D_{k'} \in \mathbb{D}_{k'}} F_{k'} \circledast D_{k'}}_{dynamic\ conv} \tag{6.4}$$

Next, $F_{k'}$ is used as the input for the second-tier set-level collective computation (Figure 6.2 (b)). Likewise, we consider the same three types of operations:

$$F_k = o_f^{k' \to k}(F_{k'}), \tag{6.5}$$

$$A_k = o_a^{k' \to k}(F_{k'}), \tag{6.6}$$

$$\mathbb{D}_k = \left\{ o_d^{k' \to k}(F_{k'}) \right\}, \tag{6.7}$$

and form the inner node F_k via further feature self-calibration and transformation as:

$$F_k = \underbrace{F_k}_{feature} \oplus \underbrace{(F_k \odot A_k)}_{attention} \oplus \underbrace{\sum_{D_k \in \mathbb{D}_k} F_k \circledast D_k}_{dynamic\ conv} \tag{6.8}$$

In doing so, our heterogeneous residual block presents a two-tier combinatorial operations structure for each inner node, resulting in a more expressive search space (see Section 6.3.3).

6.2.3 Attention Guided Search Optimisation in a Heterogeneous Search Space

To showcase the effectiveness of the proposed heterogeneous search space and operator cell, we adopt the proxy-based NAS strategy, due to the computing resource constraints and the enormous search space. This search is done by constructing a small proxy network parametrised by Θ.

Attention Guided Search. Compared with proxyless search strategy, proxy-based NAS is more efficient but relatively less optimal due to *not* directly optimising the final network architecture. This training-test discrepancy problem can be worsened when the search space provides more flexibility and combinatorial capability, such as the proposed space. To solve this obstacle, we propose attention guided search [1], which optimises the proxy network in a knowledge distillation manner by injecting an external guidance from a pre-trained teacher network into the NAS process.

Specifically, we leverage the attention transfer idea [96] that encourages a student (the proxy network in our case) to hierarchically imitate a teacher's hidden attention knowledge. Intuitively, this may benefit the search for self-calibration learning. Formally, let us denote a feature tensor at the j-th stage of the teacher and student network as $\boldsymbol{F}_T^j$ and $\boldsymbol{F}_S^j$, separately. Attention transfer is realised by imposing an alignment loss function across the two networks as:

$$\mathcal{L}_{AT} = \frac{1}{2}\sum_{j\in\mathcal{J}} \| \frac{\boldsymbol{x}_S^j}{\|\boldsymbol{x}_S^j\|_2} - \frac{\boldsymbol{x}_T^j}{\|\boldsymbol{x}_T^j\|_2} \|_2, \tag{6.9}$$

where $\boldsymbol{x}_{S/T}^j = vec(\sum_i |\boldsymbol{F}_{S/T}^j(\cdot,\cdot,i)|^2)$ is the spatial-wise accumulated feature vector. An overview of attention guided search is depicted in Figure 6.3.

Optimisation. For NAS optimisation, we adopt the DARTS method [14]. In our context, we conduct the continuous relaxation over all the possible heterogeneous operations $\mathcal{O}$ for making a continuous search space:

$$\overline{o}^{i\to j}(x) = \sum_{o\in\mathcal{O}} \frac{\exp\left(\boldsymbol{a}_o^{i\to j}\right)}{\sum_{o'\in\mathcal{O}}\exp\left(\boldsymbol{a}_{o'}^{i\to j}\right)} o(x), \tag{6.10}$$

where an architecture vector $\boldsymbol{a}_o^{i\to j} \in \mathbb{R}^{|\mathcal{O}|}$ is used for each possible connection $i \to j$. We summarise the architecture vector of all the connections as a matrix $\boldsymbol{A} = \left[\boldsymbol{a}^1, \cdots, \boldsymbol{a}^{|\mathcal{E}|}\right] \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{O}|}$. With this relaxation, we can jointly optimise the architecture parameters $\boldsymbol{A}$ and the network weights Θ in a fully gradient differentiable manner.

Equipped with the proposed attention guidance search, the search objective function is finally formulated as the following bilevel optimisation process:

[1] We would like to point out that there is a related concurrent work [158] also applies distillation method for NAS.

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}_{train}(\Theta, \boldsymbol{A}) + \lambda \mathcal{L}_{AT}(\Theta, \boldsymbol{A}), \tag{6.11}$$

$$\boldsymbol{A}^* = \arg\min_{\boldsymbol{A}} \mathcal{L}_{val}(\Theta^*, \boldsymbol{A}) + \lambda \mathcal{L}_{AT}(\Theta^*, \boldsymbol{A}), \tag{6.12}$$

where λ denotes the weighting hyper-parameter. For the first level Eq. (6.11), we learn the optimal parameters Θ^* for a given architecture $\boldsymbol{A}$ w.r.t a training objective $\mathcal{L}_{train}$ and the attention alignment loss $\mathcal{L}_{AT}$. The second level Eq. (6.12) then explores the optimal architecture $\boldsymbol{A}^*$ over the heterogeneous search space $\mathbb{A}$ w.r.t a validation objective $\mathcal{L}_{val}$ and $\mathcal{L}_{AT}$. For image classification, $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ usually take the cross-entropy loss function.

Search Outcome. Once the above alternated optimisation is done, we derive an amenable cell architecture with heterogeneous operators. In practice, for each heterogeneous computing block we retain the top-2 strongest incoming operations with at least one feature transformation operation for the first-tier (Figure 6.2(a)), and the top-1 strongest operation for the second-tier (Figure 6.2(b)).

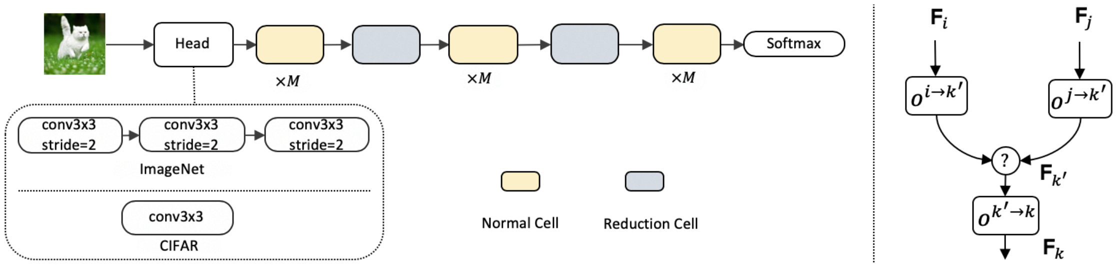


Figure 6.4: (**Left**) The overall model architecture for CIFAR-10 and ImageNet, consisting of repeated Normal Cells and Reduction Cells. M is the stacking choice for the number of Normal Cells. Each cell contains 4 blocks. (**Right**) An example of two-tier block construction in cell: Each block takes two input features (F_i, F_j) from previous nodes; The operator (?) in a block is determined by the choices of two operations ($o^{i\to k'}$, $o^{j\to k'}$) in the first-tier; An extra operation ($o^{k'\to k}$) is selected in the second-tier.

6.3 Experiments

Datasets. We evaluated the proposed NOS method on image classification using three common datasets. *CIFAR10/100:* Both CIFAR10 and CIFAR100 have 50K/10K train/test RGB images of size $32 \times 32 \times 3$, categorised into 10 and 100 classes, respectively [140]. *ImageNet:* We used the ILSVRC2012 version for large-scale image classification evaluation, containing 1.28M training images and 50K validation samples from 1,000 object classes [2].

Model	Type	Kernels	CIFAR10		CIFAR100		FLOPS(M)	#Params(MB)
			Top-1(%)	Top-5(%)	Top-1(%)	Top-5(%)		
ResNet-18	-	-	4.95	0.22	23.61	7.16	555.42	11.17
+ Dynamic	Normal	3	4.63 ↑	0.13 ↑	22.63 ↑	6.44 ↑	+ 3.85	+ 0.03
		5	4.78 ↑	0.14 ↑	23.45 ↑	6.82 ↑	+ 7.62	+ 0.04
	Dilated	3	4.97 ↓	0.23 ↓	24.00 ↓	7.28 ↓	+ 3.85	+ 0.03
		5	4.92 ↑	0.17 ↑	23.75 ↓	7.20 ↓	+ 7.62	+ 0.04
+ Attention	Spatial		4.79 ↑	0.16 ↑	23.51 ↑	7.04 ↑	+ 1.08	+ 0.01
	Channel		4.83 ↑	0.19 ↑	23.20 ↑	6.89 ↑	+ 0.40	+ 0.15

↑ Better than the baseline. ↓ Worse than the baseline.

Table 6.1: Evaluating the feature self-calibration operations on CIFAR10 and CIFAR100.

We first conduct preliminary experiments on CIFAR10/100 to select the heterogeneous primitive operations $\mathcal{O}$. To test the efficacy and transferability of NOS, we search the cell structures on CIFAR10 only, and compare the performance with existing methods on CIFAR10/100 and ImageNet.

6.3.1 Details of Training Configurations

We introduce the details of training configurations as follows.

ResNet-18 and PyramidNet-110. We trained these models for 300 epochs with batch size 32. The learning rate was initialised as 0.025, which was decayed by 10 every 30 epochs. The standard SGD optimiser with momentum of 0.9 was employed. We set a weight decay value of 1×10^{-4} to avoid overfitting. Other additional enhancements were not involved except the standard data augmentations.

Cell Search. For network parameters Θ of proxy network, we used SGD with an initial learning rate 0.025 and set the momentum value as 0.9. This learning rate was decayed to 0 with a cosine scheduler. A weight decay value of 3×10^{-4} was imposed to avoid over-fitting. For learning architecture matrix $\boldsymbol{A}$, we used the Adam optimiser with a fixed learning rate value 6×10^{-4} and set the weight decay to 1×10^{-3}.

Cell Evaluation. The evaluation network was trained from scratch directly for 600 epochs with batch size 128. Note that, the attention transfer was **not** involved for training. We set the weight decay values for CIFAR-10 and CIFAR-100 to 3×10^{-4} and 5×10^{-4} individually. The standard SGD optimiser with a momentum of 0.9 was applied. The initial learning rate was 0.25, decayed to 0 with a cosine scheduler. Following existing works [14, 11, 10, 13], we performed two additional enhancements: the cutout regularisation [150] with length 16 and the drop-path [151] of probability 0.3.

ImageNet. We trained the evaluation model for ImageNet using SGD optimiser for 300 epochs with batch size 512. We initialised the learning rate as 0.25 and reduced it to 0 by a linear scheduler. Learning rate warmup [152] was applied for the first 5 epochs to deal with the large batch size and learning rate.

6.3.2 Study of Feature Self-Calibration Operations

We conducted a controlled experiment to test the introduced self-calibration operations on CIFAR-10 and CIFAR-100. Specifically, for the proposed *dynamic convolutions*, we considered both normal and dilated convolutions and two kernel sizes (3 × 3 and 5 × 5). We adopted the channel-wise and spatial-wise *attention learning*. For the baseline model, we used ResNet-18 [3] with 4 stages in the backbone. To build a model with self-calibration, we added each self-calibration operation at the stages 1, 2, 3 of ResNet-18, respectively. For fair comparison, we trained each model in the same setting (see Section 6.3.1). In 6.1, we summarised the model parameters and FLOPs in addition to the test set performance (error rates). We observed that: (1) Both attention operations and our normal dynamic convolutions outperform the baseline consistently; (2) Adding dilated dynamic convolutions causes performance drop in most cases. We hence exclude it from the candidate set; (3) Very marginal FLOPs and parameters increase from these self-calibration operations over the baseline, suggesting their high cost-effectiveness.

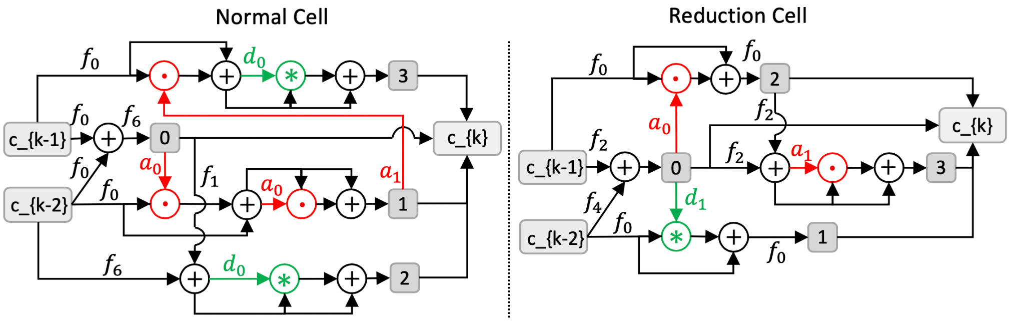


Figure 6.5: Normal cell and reduction cell searched on CIFAR-10. f_0: sep_conv_3x3, f_1: sep_conv_5x5, f_2: dil_conv_3x3, f_4: max_pooling, f_6: identity, a_0: spatial_attention, a_1: channel_attention, d_0: dynamic_conv_3x3, d_1: dynamic_conv_5x5.

Architecture	**Error (%)**		**Params**	**Search Cost**	
	CIFAR10	**CIFAR100**	**(M)**	**GPUs**	**Days**
PyramidNet [159]*	3.92	20.11	2.5	-	-
ENAS [11]	2.89	-	4.6	1	0.5
DARTS(1st) [14]	3.00±0.14	-	3.3	1	1.5
DARTS(2nd) [14]	2.76±0.09	17.54	3.3	1	4.0
SNAS (moderate) [23]	2.85±0.02	-	2.8	1	1.5
GHN [144]	2.84±0.07	-	5.7	1	0.84
GDAS [145]	2.93	18.38	3.4	1	0.84
BayesNAS [146]	2.81±0.04	-	3.4	1	0.2
ASNG [147]	2.83±0.14	-	3.9	1	**0.11**
Random Baseline‡	3.85	21.66	2.4	-	-
NOS (best)	**2.53**	**16.21**	2.6	1	0.35
NOS (average)	**2.67±0.06**	**16.72±0.24**	2.6	1	0.35

* The teacher model. ‡ Best architecture among 30 random samples.

Table 6.2: Comparisons with the state-of-the-art architectures obtained by proxy-based NAS methods on CIFAR10 and CIFAR100.

6.3.3 Cell Search

Search Space. Following the setup of existing methods [13, 14, 15, 147], we searched the convolutional architectures on CIFAR10. We constructed a small proxy network with 8 heterogeneous operator cells, and two reduction cells at 1/3 and 2/3 of the total network depth for feature shape reduction. Figure 6.4 illustrates the general model architecture. As found out above, the heterogeneous primitive operation set $\mathcal{O}$ contains 11 operations in total: $|\mathcal{O}_f| = 7$ feature transformation operations, $|\mathcal{O}_a| = 2$ attention learning operations, $|\mathcal{O}_d| = 2$ dynamic convolutions, respectively. We constructed the proposed heterogeneous operator cell ($\mathcal{G} = (\mathcal{V}, \mathcal{E})$) with $|\mathcal{V}| = 7$ nodes (2 input nodes, 4 inner nodes and 1 output node). So, all 4 heterogeneous residual blocks contain $|\mathcal{E}| = 18$ edges in total (14 first-tier connections and 4 second-tier connections). To derive the final cell architecture, we kept 2 first-tier connections and 1 second-tier connection for each block. As a result, there is a total number of $\prod_{n=1}^{4} \frac{(n+1)n}{2} \times 11^3 \approx 10^{14}$ possible choices, 5 orders of magnitude larger than the conventional size of $\prod_{n=1}^{4} \frac{(n+1)n}{2} \times 7^2 \approx 10^9$ as in [14, 145, 23].

Training. Following the setup of existing methods [13, 14, 15, 147], we searched the convolutional architectures on CIFAR10. We constructed a small proxy network with 8 heterogeneous operator cells, and two reduction cells at 1/3 and 2/3 of the total network depth for feature shape reduction (see Section 6.3.1). We used 25K images split from the training set for validation. We randomly initialised the architecture parameters $\boldsymbol{A} \in \mathbb{R}^{18 \times 11}$ in the normal distribution. We used a pre-trained PyramidNet-110 (bottleneck, $\alpha = 84$) [159] as the teacher model. We set the weight $\lambda = 10^3$ for attention guidance loss $\mathcal{L}_{AT}$. After 25 epochs of training on the proxy network, we derived the final heterogeneous operator cells from the architecture matrix $\boldsymbol{A}$. See Section 6.3.1 for more configurations for training the proxy and teacher networks.

The search on CIFAR10 took only 8.4 hours using a single NVIDIA Tesla V100 GPU. The searched heterogeneous operator cells by NOS is shown in Figure 6.5, in which the self-calibration operators $\odot$ and $\circledast$ appear in both first-tier and second-tier. For example, there are two *attention* operations in first-tier and two *dynamic convolutions* in second-tier in the normal cell.

6.3.4 Further Analysis

We evaluated attention guided search (AGS) on CIFAR10/100 by comparing a NOS variant without attention transfer loss. The same training setting was used (Section 6.3.1). We used a pre-trained PyramidNet-110 as the teacher. Table 6.3 shows that learning with attention guidance can significantly benefit the NOS search process.

AGS	Test Error (%)	
	CIFAR10	CIFAR100
w/o	3.44	18.80
w/	2.53	16.21

Table 6.3: Testing attention guided search (AGS).

Distillation Effect. We examined the effect of knowledge distillation in the proposed Attention Guided Search (AGS). We conducted this analysis on CIFAR10. In this evaluation, we compared three methods: (1) Vanilla Search: Using the original DARTS search method; (2) Distillation Only: Using the attention transfer loss for training the proxy network only; (3) AGS: The proposed method (full). We tracked the model performance on both training (train) and validation (val) data sets. Figure 6.6 shows that (i) knowledge distillation brings a positive performance gain over the vanilla DARTS and (ii) using attention guidance for the network search can further improve the searched architecture. This suggests that distillation is effective to alleviate the architecture training-test discrepancy issue involved in the proxy-based NAS.

Space Generality. We tested the general effect of the proposed Attention Guided Search (AGS) using the original DARTS search space ($\mathcal{O}_f$ + zero operation) on CIFAR10. During the search process, we followed the same settings as DARTS with the first-order optimisation. We

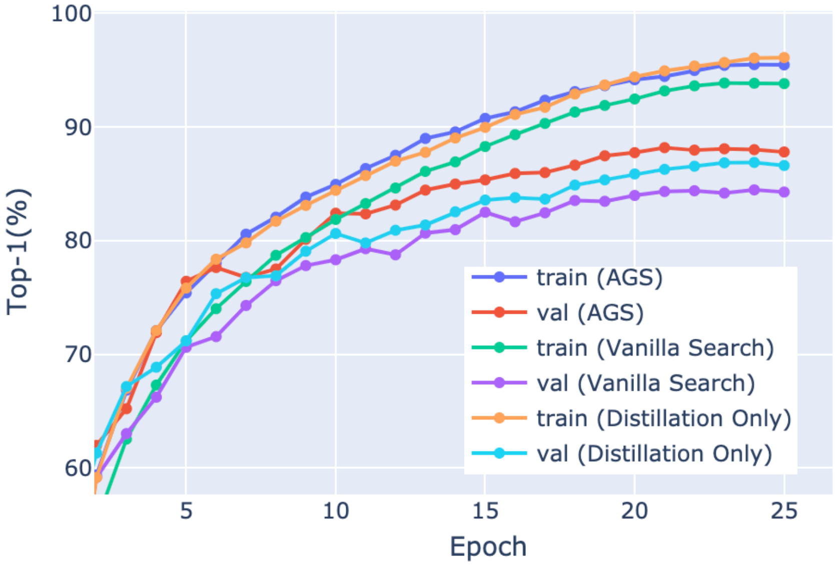


Figure 6.6: The train and val set accuracies on CIFAR10.

obtained the error rates: 3.00 ± 0.14 (DARTS) *vs.* 2.92 ± 0.05 (DARTS + AGS). This suggests a general efficacy of AGS over different search spaces.

6.3.5 Architecture Evaluation

We evaluate the searched .architectures on both CIFAR and ImageNet benchmarks.

CIFAR. To measure the final image classification performance of the searched heterogeneous operator cells on CIFAR10 and CIFAR100, we created an evaluation network with 20 cells, 36 initial channels, and an auxiliary tower with loss weight 0.4. See Section 6.3.1 for more configurations for training the evaluation network. Due to high variance of results on CIFAR, we conducted 10 independent runs and reported both the best and average results. We summarised the results of NOS and the state-of-the-art proxy-based NAS methods[2] in Table 6.2. The comparisons show that: **(1)** NOS achieves the best result on CIFAR10, whilst enjoying the smallest model parameters (only 2.6M). It shows the significant cost-effectiveness and compactness advantages of our method. **(2)** Despite a significantly larger search space (10^{14} *vs.* 10^{9} in [14, 147, 23, 145, 147]), NOS shows high cost-effectiveness in computing cost (only 0.35 GPU day). **(3)** NOS achieves the best result on CIFAR100 by directly transferring the CIFAR10 searched network, significantly outperforming DARTS [14] and GDAS [145]. This challenging

[2] In our evaluation context, we primarily aim to accurately evaluate the effect of search space. To this end, we selectively compared with a set of more related state-of-the-art NAS methods, rather than exhaustively. For instance, we excluded ProxylessNAS [18] due to that it uses a different search strategy with a *supernet* (orthogonal to the search space factor).

cross-dataset test indicates a superior transferability of the network searched by NOS.

ImageNet. To evaluate the transferability of architecture discovered by NOS on large scale ImageNet, we used the mobile setting same as in [14, 145, 23], where the number of multiply-add operations is restricted to be less than 600M at the input size of 224×224. Specifically, we constructed an evaluation network with 14 cells and 48 initial channels. An auxiliary tower with loss weight 0.4 was also applied. See Section 6.3.1 for more training details. Table 6.4 shows the ImageNet results in the mobile setting. Notably, compared to other state-of-the-art proxy-based NAS using gradient optimisation (GHN [144], DARTS [14], SNAS [23], GDAS [145] and BayesNAS [146]), the network searched by NOS on CIFAR10 can be successfully transferred. Also, NOS discovers a cell structure that performs better with higher efficiency (only 440M FLOPs).

Architecture	**Test Err. (%)**		**Params**	$\times+$	**Search Cost**
	top-1	**top-5**	**(M)**	**(M)**	**(GPU-days)**
GHN	27.0	8.7	6.1	569	0.84
DARTS	26.7	8.7	4.7	574	4.0
SNAS	27.3	9.2	4.3	522	1.5
GDAS	26.0	8.5	5.3	581	0.84
BayesNAS	26.5	8.9	**3.9**	-	**0.2**
NOS	**25.8**	**8.1**	4.0	**440**	0.35

Table 6.4: Comparisons with the state-of-the-art proxy-based architectures on ImageNet-mobile.

6.4 Summary

In this chapter, we presented Neural Operator Search (NOS), characterised by a heterogeneous search space for neural architecture search (NAS). Specifically, NOS further introduces dynamic convolution and attention learning operations on top of the conventional feature transformation operations. This proposed search space expansion enables NAS to discover more expressive and previously undiscovered architectures, significantly expanding the search horizon and enriching the possible search outcomes. Moreover, we introduced a heterogeneous operator cell to integrate these different operations synergistically. To facilitate the learning process, we further proposed an attention guided search mechanism in a distillation manner. Extensive evaluations have validated the superiority of our method over a wide range of state-of-the-art NAS models on the standard image classification tasks.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This thesis has studied a wide range of approaches to designing scalable deep learning architectures for several computer vision tasks by considering the underlying task characteristics for more efficient and powerful deep models. In particular, the primary aims of this thesis in research of scalable deep learning architectures are (i) maximise the cost-effectiveness for scalable model deployments; (ii) reduce the design cost of scalable deep architectures; (iii) improve the generalisation of scalable deep architectures. We concentrate on three critical computer vision tasks: person re-identification, person search, and image recognition. These tasks are inherently challenging due to intrinsic visual ambiguities and high demand for scalability requirements. Therefore, designing scalable deep learning architectures for these challenging tasks are studied by considering the underlying task characteristics for more efficient and powerful deep models. Specifically,

1. We have designed scalable deep learning architectures to address (i) in Chapter 3 and Chapter 4. Particularly, Chapter 3 formulates a novel harmonious attention network (HAN) framework to jointly learn soft pixel attention and hard region attention alongside simultaneous deep feature representation learning. This enables more discriminative re-id matching by efficient networks with more scalable model inference and feature matching. On the other hand, Chapter 4 proposes a Hierarchical Distillation Learning (HDL) approach, to comprehensively distil the knowledge of a strong teacher model with strong learning ca-

pability to a lightweight student model with weak learning capability. Despite differences in model design details, both HAN and HDL are scalable to large scale re-id deployment scenarios with the need of processing a large amount of surveillance video data, due to the lengthy inference process with high computing costs. Also, we propose a novel Neural Operator Search (NOS) method, which leverages attention guided search for facilitating the search process over a vast search space more efficiently and more effectively.

2. We have automatic searched scalable deep learning architectures to address (ii) in Chapter 5 and Chapter 6. Specifically, Chapter 3 proposes a novel method Neural Graph Embedding (NGE) to address the limitation of the state-of-the-art methods in mining network topology knowledge. We jointly model the graphical topology of network architecture and performing the network search in continuous representation space. Meanwhile, Chapter 6 presents a novel heterogeneous search space for NAS, which contains richer primitive operations including both conventional feature transformations and newly introduced feature self-calibration. Also, we propose a novel Neural Operator Search (NOS) method dedicated for NAS, which leverages attention guided search for facilitating the search process over a vast search space more efficiently and more effectively. With extensive experiments, we demonstrate that both NGE and NOS allows to automatic design scalable deep architectures within one GPU day.

3. To address (iii), we conduct extensive evaluations to verify the generalisation of our manually designed deep learning architectures on benchmark datasets. In particular, in Chapter 3, we validate the cost-effectiveness superiority of the proposed HAN approach for person re-id against a wide variety of state-of-the-art methods on four large benchmark datasets: CUHK03, Market-1501, DukeMTMC, and MSMT17. Chapter 4 shows model cost-effectiveness and performance advantages of our HDL over the state-of-the-art alternative approaches on three person search benchmarks: CUHK-SYSU, DukeMTMC-PS, and PRW . Also, our automatic searched deep learning architectures by Neural Graph Embedding (NGE) in Chapter 5 and Neural Operator Search (NOS) in Chapter 6, can not only achieves highly competitive accuracy performance on the searched dataset (CIFAR-10) but also generalise well on CIFAR-100 and ImageNet datasets. Moreover, the neural architecture discovered on CIFAR-10 by NGE can be readily transferred to the more challenging semantic segmentation task (PASCAL VOC 2012).

In this thesis, although studied on there different problems as we presented in separate chapters, scalable deep learning architectures proposed in Chapter 3, 4, 5 and 6 would also have potentials and benefits for dealing with other relevant tasks in computer vision. More discussions about future research directions and work are detailed below.

7.2 Future Work

The potential research directions for future work beyond the proposed methods are summarised as follows to end this thesis. Despite remarkable successes achieved by manual designed scalable deep learning architectures in Chapter 3 and 4, our main concentration is to extend two scalable deep learning architecture search methods: Neural Graph Embedding (NGE) in Chapter 5 and Neural Operator Search (NOS) in Chapter 6. This is due to the fact that, compared with manual designs, automatic search methods requires much less human expertise and interventions, which shows much greater potential in speeding up the designing process of scalable deep architectures.

NGE is capable of modelling the topology knowledge of the network architecture by recursive message propagation in a constructed continuous search space. In particular, NGE elegantly enables integrating the Graph Convolutional Network (GCN) [132] with the existing gradient-based paradigm. Therefore, it models the topology of the network architecture, facilitating the NAS design with favourable search efficiency. However, existing gradient-based methods cannot properly preserve the graph information when projecting a neural architecture into a continuous space. This, in the end, would cause inaccuracy and limited representation capability in the mapped space. Moreover, existing approaches usually explore only a very limited inner-cell search space. The construction small-scale cell space results in representation limitation and poor scalability. Thus, it is valuable to study a more advanced graph embedding method to enable quick search of more sophisticated neural architectures while preserving graph information.

NOS presents a heterogeneous search space for neural architecture search (NAS), introducing dynamic convolution and attention learning operations on top of the conventional feature transformation operations. Therefore, it enables NAS to discover more expressive and previously undiscovered architectures, significantly expanding the search horizon and enriching the possible search outcomes. These dynamic convolutions and attention learning operations greatly enhance the learning ability of conventional deep models by enabling a neural network to focus more on relevant elements of the input than on irrelevant parts. Thus, our proposed heterogeneous search space in NOS that contains these advanced operations could be naturally applied in searching scalable deep learning architectures for a broad range of sophisticated computer vision tasks, such as visual question answering, image-text retrieval, and so on.

Bibliography

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[6] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2017.

[7] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[8] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L. Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[9] Vladimir Nekrasov, Hao Chen, Chunhua Shen, and Ian Reid. Fast neural architecture search of compact semantic segmentation models via auxiliary cells. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018.

[11] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *International Conference on Machine learning*, 2018.

[12] Hanxiao Liu, Karen Simonyan, Oriol Vinyals, Chrisantha Fernando, and Koray Kavukcuoglu. Hierarchical representations for efficient architecture search. In *International Conference on Learning Representations*, 2018.

[13] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *AAAI Conference on Artificial Intelligence*, 2019.

[14] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[15] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *European Conference on Computer Vision*, pages 19–34, 2018.

[16] Jorge Sánchez and Florent Perronnin. High-dimensional signature compression for large-scale image classification. pages 1665–1672. IEEE, 2011.

[17] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[18] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2019.

[19] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mo-

bile. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[20] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019.

[21] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient architecture search by network transformation. In *AAAI Conference on Artificial Intelligence*, 2018.

[22] Han Cai, Jiacheng Yang, Weinan Zhang, Song Han, and Yong Yu. Path-level network transformation for efficient architecture search. In *International Conference on Machine learning*, 2018.

[23] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. Snas: stochastic neural architecture search. In *International Conference on Learning Representations*, 2019.

[24] Renqian Luo, Fei Tian, Tao Qin, Enhong Chen, and Tie-Yan Liu. Neural architecture optimization. In *Advances in Neural Information Processing Systems*, pages 7816–7827, 2018.

[25] Christian Sciuto, Kaicheng Yu, Martin Jaggi, Claudiu Musat, and Mathieu Salzmann. Evaluating the search phase of neural architecture search. *arXiv e-print*, 2019.

[26] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[27] Luca Bertinetto, João F Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *Advances in Neural Information Processing Systems*, pages 523–531, 2016.

[28] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[29] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pages 667–675, 2016.

[30] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*, 2019.

[31] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. *arXiv e-print*, 2019.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[33] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. Citypersons: A diverse dataset for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2017.

[34] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Reidentification by relative distance comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):653–668, March 2013.

[35] Shengcai Liao, Yang Hu, Xiangyu Zhu, and Stan Z Li. Person re-identification by local maximal occurrence representation and metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[36] Tetsu Matsukawa, Takahiro Okabe, Einoshin Suzuki, and Yoichi Sato. Hierarchical gaussian descriptor for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[37] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a discriminative null space for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[38] Hanxiao Wang, Xiatian Zhu, Shaogang Gong, and Tao Xiang. Person re-identification in identity regression space. *International Journal of Computer Vision*, pages 1–23, 2018.

[39] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[40] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, 2015.

[41] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[42] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Workshop of European Conference on Computer Vision*, 2016.

[43] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*, 2017.

[44] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 3, 2017.

[45] Xuelin Qian, Yanwei Fu, Yu-Gang Jiang, Tao Xiang, and Xiangyang Xue. Multi-scale deep learning architectures for person re-identification. In *IEEE International Conference on Computer Vision*, 2017.

[46] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. A multi-task deep network for person re-identification. In *AAAI Conference on Artificial Intelligence*, 2017.

[47] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[48] Faqiang Wang, Wangmeng Zuo, Liang Lin, David Zhang, and Lei Zhang. Joint learning of single-image and cross-image representations for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[49] Wei Li, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep joint learning of multi-loss classification. In *International Joint Conference of Artificial Intelligence*, 2017.

[50] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Person re-identification by deep learning multi-scale representations. In *Workshop of IEEE International Conference on Computer Vision*, 2017.

[51] Mahdi M. Kalayeh, Emrah Basaran, Muhittin Gkmen, Mustafa E. Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[52] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[53] Xiaobin Chang, Timothy M. Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[54] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[55] Zhun Zhong, Liang Zheng, Shaozi Li, and Yi Yang. Generalizing a person retrieval model hetero-and homogeneously. In *European Conference on Computer Vision*, pages 172–188, 2018.

[56] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised person re-identification by deep learning tracklet association. In *European Conference on Computer Vision*, 2018.

[57] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Deep association learning for unsupervised video person re-identification. In *British Machine Vision Conference*, 2018.

[58] Elyor Kodirov, Tao Xiang, and Shaogang Gong. Dictionary learning with iterative laplacian regularisation for unsupervised person re-identification. In *British Machine Vision Conference*, 2015.

[59] Peixi Peng, Tao Xiang, Yaowei Wang, Massimiliano Pontil, Shaogang Gong, Tiejun Huang, and Yonghong Tian. Unsupervised cross-dataset transfer learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[60] Hanxiao Wang, Xiatian Zhu, Tao Xiang, and Shaogang Gong. Towards unsupervised open-set person re-identification. In *IEEE International Conference on Image Processing*, 2016.

[61] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *IEEE International Conference on Computer Vision*, 2017.

[62] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv e-print*, 2017.

[63] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[64] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[65] Misha Denil, Babak Shakibi, Laurent Dinh, Nando De Freitas, et al. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems*, pages 2148–2156, 2013.

[66] Xiatian Zhu, Botong Wu, Dongcheng Huang, and Wei-Shi Zheng. Fast open-world person re-identification. *IEEE Transactions on Image Processing*, 27(5):2286–2300, 2018.

[67] Ruimao Zhang, Liang Lin, Rui Zhang, Wangmeng Zuo, and Lei Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, 2015.

[68] Fuqing Zhu, Xiangwei Kong, Liang Zheng, Haiyan Fu, and Qi Tian. Part-based deep hashing for large-scale person re-identification. *IEEE Transactions on Image Processing*, 26(10):4806–4817, 2017.

[69] Yan Wang, Lequn Wang, Yurong You, Xu Zou, Vincent Chen, Serena Li, Gao Huang, Bharath Hariharan, and Kilian Q Weinberger. Resource aware person re-identification

across multiple resolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8042–8051, 2018.

[70] Yang Shen, Weiyao Lin, Junchi Yan, Mingliang Xu, Jianxin Wu, and Jingdong Wang. Person re-identification with correspondence structure learning. In *IEEE International Conference on Computer Vision*, 2015.

[71] Wei-Shi Zheng, Xiang Li, Tao Xiang, Shengcai Liao, Jianhuang Lai, and Shaogang Gong. Partial person re-identification. In *IEEE International Conference on Computer Vision*, pages 4678–4686, 2015.

[72] Hanxiao Wang, Shaogang Gong, and Tao Xiang. Unsupervised learning of generative topic saliency for person re-identification. In *British Machine Vision Conference*, 2014.

[73] Rui Zhao, Wanli Ouyang, and Xiaogang Wang. Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[74] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 384–393, 2017.

[75] Liming Zhao, Xi Li, Jingdong Wang, and Yueting Zhuang. Deeply-learned part-aligned representations for person re-identification. In *IEEE International Conference on Computer Vision*, 2017.

[76] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *ICCV*, 2017.

[77] Xu Lan, Hanxiao Wang, Shaogang Gong, and Xiatian Zhu. Deep reinforcement learning attention selection for person re-identification. In *British Machine Vision Conference*, 2017.

[78] Jing Xu, Rui Zhao, Feng Zhu, Huaming Wang, and Wanli Ouyang. Attention-aware compositional network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[79] Cheng Wang, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Mancs: A

multi-task attentional network with curriculum sampling for person re-identification. In *European Conference on Computer Vision*, September 2018.

[80] Xuelin Qian, Yanwei Fu, Tao Xiang, Wenxuan Wang, Jie Qiu, Yang Wu, Yu-Gang Jiang, and Xiangyang Xue. Pose-normalized image generation for person re-identification. In *European Conference on Computer Vision*, September 2018.

[81] Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoung Mu Lee. Part-aligned bilinear representations for person re-identification. In *European Conference on Computer Vision*, September 2018.

[82] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[83] Xihui Liu, Haiyu Zhao, Maoqing Tian, Lu Sheng, Jing Shao, Shuai Yi, Junjie Yan, and Xiaogang Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. In *IEEE International Conference on Computer Vision*, 2017.

[84] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3415–3424, 2017.

[85] Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Chandraker, Yi Yang, and Qi Tian. Person re-identification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1367–1376, 2017.

[86] Xiaojun Chang, Po-Yao Huang, Yi-Dong Shen, Xiaodan Liang, Yi Yang, and Alexander G Hauptmann. Rcaa: Relational context-aware agents for person search. In *European Conference on Computer Vision*, pages 84–100, 2018.

[87] Jimin Xiao, Yanchun Xie, Tammam Tillo, Kaizhu Huang, Yunchao Wei, and Jiashi Feng. Ian: the individual aggregation network for person search. *Pattern Recognition*, 87:332–340, 2019.

[88] Hao Liu, Jiashi Feng, Zequn Jie, Karlekar Jayashree, Bo Zhao, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. Neural person search machines. In *IEEE International Conference on Computer Vision*, pages 493–501, 2017.

[89] Xu Lan, Xiatian Zhu, and Shaogang Gong. Person search by multi-scale matching. In *European Conference on Computer Vision*, pages 536–552, 2018.

[90] Di Chen, Shanshan Zhang, Wanli Ouyang, Jian Yang, and Ying Tai. Person search via a mask-guided two-stream cnn model. In *European Conference on Computer Vision*, pages 734–750, 2018.

[91] Bharti Munjal, Sikandar Amin, Federico Tombari, and Fabio Galasso. Query-guided end-to-end person search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 811–820, 2019.

[92] Yichao Yan, Qiang Zhang, Bingbing Ni, Wendong Zhang, Minghao Xu, and Xiaokang Yang. Learning context graph for person search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2158–2167, 2019.

[93] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.

[94] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, 2015.

[95] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[96] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017.

[97] David Navon. Forest before trees: The precedence of global features in visual perception. *Cognitive Psychology*, 9(3):353–383, 1977.

[98] Antonio Torralba, Aude Oliva, Monica S Castelhano, and John M Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review*, 113(4):766, 2006.

[99] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[100] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, 2007.

[101] Yantao Shen, Hongsheng Li, Tong Xiao, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Deep group-shuffling random walk for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[102] Dapeng Chen, Dan Xu, Hongsheng Li, Nicu Sebe, and Xiaogang Wang. Group consistent similarity learning via deep crf for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[103] Yicheng Wang, Zhenzhong Chen, Feng Wu, and Gang Wang. Person re-identification with cascaded pairwise convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[104] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. Gated siamese convolutional neural network architecture for human re-identification. In *European Conference on Computer Vision*, 2016.

[105] Arulkumar Subramaniam, Moitreya Chatterjee, and Anurag Mittal. Deep neural networks with inexact matching for person re-identification. In *Advances in Neural Information Processing Systems*, 2016.

[106] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[107] Shimon Edelman. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21(04):449–467, 1998.

[108] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine learning*, pages 448–456, 2015.

[109] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[110] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. 2018.

[111] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[112] Simone Vossel, Joy J Geng, and Gereon R Fink. Dorsal and ventral attention systems: distinct neural circuits but collaborative roles. *The Neuroscientist*, 20(2):150–159, 2014.

[113] Laurent Sifre and PS Mallat. *Rigid-motion scattering for image classification*. PhD thesis, Citeseer, 2014.

[114] Zhun Zhong, Liang Zheng, Donglin Cao, and Shaozi Li. Re-ranking person re-identification with k-reciprocal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[115] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. 2017.

[116] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[117] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv*, 2017.

[118] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*, 2017.

[119] Ying-Cong Chen, Xiatian Zhu, Wei-Shi Zheng, and Jian-Huang Lai. Person re-identification by camera correlation aware feature augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[120] Hao Liu, Jiashi Feng, Meibin Qi, Jianguo Jiang, and Shuicheng Yan. End-to-end comparative attention networks for person re-identification. *IEEE Transactions on Image Processing*, 2017.

[121] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. In *IEEE International Conference on Computer Vision*, 2017.

[122] Longhui Wei, Shiliang Zhang, Hantao Yao, Wen Gao, and Qi Tian. Glad: Global-local-alignment descriptor for pedestrian retrieval. In *ACM International Conference on Multimedia*, pages 420–428, 2017.

[123] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[124] Jinxian Liu, Bingbing Ni, Yichao Yan, Peng Zhou, Shuo Cheng, and Jianguo Hu. Pose transferrable person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[125] Houjing Huang, Dangwei Li, Zhang Zhang, Xiaotang Chen, and Kaiqi Huang. Adversarially occluded samples for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[126] Xiaobin Chang, Timothy M. Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[127] Zhun Zhong, Liang Zheng, Zhedong Zheng, Shaozi Li, and Yi Yang. Camera style adaptation for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[128] Sarfraz M. Saquib, Arne Schumann, Andreas Eberle, and Rainer Stiefelhagen. A pose-sensitive embedding for person re-identification with expanded cross neighborhood re-ranking. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2018.

[129] Rui Yu, Zhiyong Dou, Song Bai, Zhaoxiang Zhang, Yongchao Xu, and Xiang Bai. Hard-aware point-to-set deep metric for person re-identification. In *European Conference on Computer Vision*, September 2018.

[130] Yantao Shen, Hongsheng Li, Shuai Yi, Dapeng Chen, and Xiaogang Wang. Person re-identification with deep similarity-guided graph neural network. In *European Conference on Computer Vision*, September 2018.

[131] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[132] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems*, 2018.

[133] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[134] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.

[135] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[136] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[137] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: bottleneck attention module. In *British Machine Vision Conference*, 2018.

[138] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[139] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[140] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[141] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[142] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IEEE International Conference on Computer Vision*, pages 991–998. IEEE, 2011.

[143] Yukang Chen, Gaofeng Meng, Qian Zhang, Shiming Xiang, Chang Huang, Lisen Mu, and Xinggang Wang. Renas: Reinforced evolutionary neural architecture search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4787–4796, 2019.

[144] Chris Zhang, Mengye Ren, and Raquel Urtasun. Graph hypernetworks for neural architecture search. In *International Conference on Learning Representations*, 2019.

[145] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1761–1770, 2019.

[146] Hongpeng Zhou, Minghao Yang, Jun Wang, and Wei Pan. Bayesnas: A bayesian approach for neural architecture search. In *International Conference on Machine learning*, 2019.

[147] Youhei Akimoto, Shinichi Shirakawa, Nozomu Yoshinari, Kento Uchida, Shota Saito, and Kouhei Nishida. Adaptive stochastic natural gradient method for one-shot neural architecture search. In *International Conference on Machine learning*, 2019.

[148] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[149] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *European Conference on Computer Vision*, pages 116–131, 2018.

[150] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv e-print*, 2017.

[151] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. 2017.

[152] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv e-print*, 2017.

[153] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv e-print*, 2017.

[154] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[155] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[156] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv e-print*, 2019.

[157] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 9401–9411, 2018.

[158] Changlin Li, Jiefeng Peng, Liuchun Yuan, Guangrun Wang, Xiaodan Liang, Liang Lin, and Xiaojun Chang. Block-wisely supervised neural architecture search with knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1989–1998, 2020.

[159] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5927–5935, 2017.