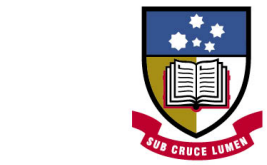


THE UNIVERSITY
of ADELAIDE

Towards Efficient Deep Neural Networks with Applications to Visual Recognition

Bohan Zhuang

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
The Unversity of Adelaide

March 2018

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Acknowledgments

Who do you want to thank? First and foremost, I would like to thank my supervisors Chunhua Shen and Ian Reid. Without their kindly and scrupulous guidance, I would not have finished my Ph.D thesis smoothly. It's my great proud for being their students. Whenever I encounter difficulties, they can provide unwavering support and effective advices to help me solve them.

I also want to thank my talented and respected postdoc collaborators who actively share their valuable knowledge with me. I really miss the impressive discussing time with them. I would like to thank Dr. Lingqiao Liu for sharing and deriving new ideas and methodologies with me, and for revising paper drafts. I also would like to thank worshiping Dr. Guosheng Lin for patiently explaining academic issues to me. I also want to thank Dr. Qi Wu and Dr. Mingkui Tan for assisting me with tackling challenging research projects.

I further thank my friends and colleagues for accompanying me in every boring day and providing happy hours to me. They are Peter Mathews, Yuanzhouhan Cao, Ruizhi Qiao, Qichang Hu, Hui Li, Zhibin Liao, Xiang Liu, Hao Lu, Yu Chen, Xiusen Wei, Ke Xian and Tong He.

Finally, I would especially thank my family for standing with me and giving me selfless love and support.

publications

The following peer-reviewed conference contain preliminary reports of the findings in this thesis (* indicates equal contribution):

1. Bohan Zhuang, Guosheng Lin, Chunhua Shen, Ian Reid; "Fast Training of Triplet-based deep binary embedding networks"; In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

2. Bohan Zhuang*, Lingqiao Liu*, Yao Li, Chunhua Shen, Ian Reid; "Attend in groups: a weakly-supervised deep learning framework for learning from web data"; In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

3. Bohan Zhuang*, Lingqiao Liu*, Chunhua Shen, Ian Reid; " Towards Context-aware Interaction Recognition for Visual Relationship Detection"; In International Conference on Computer Vision (ICCV), 2017.

4. Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid; "HCVRD: a benchmark for large-scale Human-Centered Visual Relationship Detection"; In AAAI Conference on Artificial Intelligence (AAAI), 2018.

5. Bohan Zhuang, Chunhua Shen, Mingkui Tan, Lingqiao Liu, Ian Reid; "Towards efficient low-bitwidth convolutional neural networks", under peer review.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Formulation

The thesis focuses on the following two topics: designing energy-efficient neural networks and hashing approach to make deep learning more feasible to real applications; deep convolutional neural networks for visual recognition.

1.1.1 Efficient low-bitwidth convolutional neural networks and binary data storage

Although deep learning methods have significantly improved the performance of various applications, there are still many limitations that constrain their practicality.

The first limitation of deep learning is the large number of learnable parameters and expensive computational cost, which consumes heavy computational resources and memory. To solve this problem, substantial efforts have been made to the speed-up and compression on CNNs during training, feedforward test or both of them. Among existing methods, the category of network quantization methods attracts great attention from researches and developers. This thesis tackles the problem of training a deep convolutional neural network with both low-precision weights and low-bitwidth activations. Optimizing a low-precision network is very challenging since the training process can easily get trapped in a poor local minima, which results in substantial accuracy loss. To mitigate this problem, we propose three simple-yet-effective approaches to improve the network training. First, we propose to use

a two-stage optimization strategy to progressively find good local minima. Specifically, we propose to first optimize a net with quantized weights and then quantized activations. This is in contrast to the traditional methods which optimize them simultaneously. Second, following a similar spirit of the first method, we propose another progressive optimization approach which progressively decreases the bit-width from high-precision to low-precision during the course of training. Third, we adopt a novel learning scheme to jointly train a full-precision model alongside the low-precision one. By doing so, the full-precision model provides hints to guide the low-precision model training. Extensive experiments on various datasets (i.e., CIFAR-100 and ImageNet) show the effectiveness of the proposed methods. To highlight, using our methods to train a 4-bit precision network leads to no performance decrease in comparison with its full-precision counterpart with standard network architectures (i.e., AlexNet and ResNet-50).

Another limitation for applying deep neural networks on real applications is the data storage problem. The reason is that the dimensions of low/mid/high level feature representations in conventional deep architectures are usually very huge. For instance, one middle layer of VGG16 has the dimension of $512 \times 14 \times 14$. And the commonly used fully-connected layer representation has the dimension of 4096. To solve this problem, we propose to employ hashing methods which aim to learn a mapping (or embedding) from images to a compact binary space in which Hamming distances correspond to a ranking measure for the image retrieval task. We make use of a triplet loss because this has been shown to be most effective for ranking problems. However, training in previous works can be prohibitively expensive due to the fact that optimization is directly performed on the triplet space, where the number of possible triplets for training is cubic in the number of training examples. To address this issue, we propose to formulate high-order binary codes learning as a multi-label classification problem by explicitly separating learning into two interleaved stages. To solve the first stage, we design a large-scale high-order binary codes inference

algorithm to reduce the high-order objective to a standard binary quadratic problem such that graph cuts can be used to efficiently infer the binary codes which serve as the labels of each training datum. In the second stage we propose to map the original image to compact binary codes via carefully designed deep convolutional neural networks (CNNs) and the hashing function fitting can be solved by training binary CNN classifiers. An incremental/interleaved optimization strategy is proffered to ensure that these two steps are interactive with each other during training for better accuracy. Moreover, our method demonstrates both improved training time (by as much as two orders of magnitude) as well as producing state-of-the-art hashing for various retrieval tasks.

1.1.2 Convolutional neural networks for visual recognition

Convolutional neural networks have significantly improved a wide range of visual recognition tasks (e.g., image classification [Krizhevsky et al., 2012], image detection [Redmon et al., 2016] and image segmentation [Lin et al., 2016a; He et al., 2017]). However, there are still limitations constraining the development of visual recognition. First, data matters. To achieve promising performance on a specific task, this typically requires either recruiting a team of experts [Van Horn et al., 2015] or extensive crowd-sourcing pipelines [Berg et al., 2014] to annotate large-scale datasets. A method for recognition is then trained using these expert-annotated labels, possibly also requiring additional annotations in the form of parts, attributes, or relationships which will be quite expensive and time consuming. Web images and their labels are, in comparison, much easier to obtain. But directly training on such automatically harvested images can lead to unsatisfactory performance, because the noisy labels of Web images adversely affect the learned recognition models. To address this drawback, we propose an end-to-end weakly-supervised deep learning framework which is robust to the label noise in Web images. The proposed framework relies on two unified strategies - random grouping and attention - to effectively reduce the neg-

ative impact of noisy web image annotations. Specifically, random grouping stacks multiple images into a single training instance and thus increases the labeling accuracy at the instance level. Attention, on the other hand, suppresses the noisy signals from both incorrectly labeled images and less discriminative image regions.

Second, today's state-of-the-art perceptual models have mostly tackled detecting and recognizing individual objects in isolation. However, understanding a visual scene often goes beyond recognizing individual objects. One crucial step towards a deeper understanding of visual scenes is to recognize how objects interact with each other. If we define the context of the interaction to be the objects involved, then most current methods can be categorized as either: (i) training a single classifier on the combination of the interaction and its context; or (ii) aiming to recognize the interaction independently of its explicit context. Both methods suffer limitations: the former scales poorly with the number of combinations and fails to generalize to unseen combinations, while the latter often leads to poor interaction recognition performance due to the difficulty of designing a context-independent interaction classifier. To mitigate those drawbacks, this thesis proposes an alternative, context-aware interaction recognition framework. The key to our method is to explicitly construct an interaction classifier which combines the context, and the interaction. The context is encoded via word2vec into a semantic space, and is used to derive a classification result for the interaction. The proposed method still builds one classifier for one interaction (as per type (ii) above), but the classifier built is adaptive to context via weights which are context dependent. The benefit of using the semantic space is that it naturally leads to zero-shot generalizations in which semantically similar contexts (subject-object pairs) can be recognized as suitable contexts for an interaction, even if they were not observed in the training set. Our method also scales with the number of interaction-context pairs since our model parameters do not increase with the number of interactions. Thus our method avoids the limitation of both approaches.

1.2 Main Contribution

The main contribution of this thesis includes a number of new algorithms and analysis on the two main research focuses as introduced in the previous section. More specifically, they are:

- To address the issue of prohibitively high computational complexity in triplet-based binary code learning, we propose a new efficient and flexible framework for interactively inferring binary codes and learning the deep hash functions, using a triplet-based loss function. We show how to convert the high-order loss introduced by the triplets into a binary quadratic problem that can be optimized efficiently in the manner of [Lin et al., 2014a], using block coordinate descent with graph-cuts. To learn the mapping from images to hash codes, we design deep CNNs capable of preserving their semantic ranking information of the data. Moreover, we propose a novel incremental group-wise training approach, that interleaves finding groups of bits of the hash codes, with learning the hash functions. We show experimentally that this approach improves the quality of hash functions while retaining the advantage of efficient training.

- We propose three simple-yet-effective approaches to improve the low-bitwidth network training. First, we propose to use a two-stage optimization strategy to quantize the weights and activations separately. Second, we also progressively decrease the bit-width from high-precision to low-precision during the course of training. Third, we jointly train a full-precision model alongside the low-precision one. By doing so, the full-precision model provides hints to guide the low-precision model training.

- We propose a weakly-supervised deep learning framework which is robust to the label noise in Web images. It relies on random grouping and attention unified strategies to effectively suppress the noisy signals.

- We propose a context-aware interaction recognition framework for visual rela-

tionship detection. Different to the previous methods, the interaction classifier in our method is designed to be adaptive to its context. The benefit of using the semantic space is that it naturally leads to zero-shot generalizations in which semantically similar contexts can result in similar classifiers even if they were not observed in the training set.

- We construct a large-scale human-centric visual relationship detection dataset (HCVRD), which provides many more types of relationship annotations (nearly 10K categories) than the previous released datasets. We also propose a webly-supervised approach to solve the long-tail distribution problem in this large-scale dataset.

1.3 Thesis organization

The rest of the thesis is organized as follows: In Chapter 2, a detailed literature review on energy-efficient neural networks and data storage as well as visual recognition is given. In Chapter 3, a novel low-bitwidth network optimization approach is introduced to efficiently quantize both weights and activations to low-precision with high accuracy. In Chapter 4, an efficient hashing framework is proposed to map the original feature space to Hamming space for efficient data storage and fast search. In Chapter 5, we propose a novel noise-robust weakly-supervised framework for learning from large-scale web data. In Chapter 6, we propose a context-aware interaction recognition framework for understanding how objects interact with each other. It is a necessary step for machines to understand the real world. In Chapter 7, we further propose a large-scale human-centric visual relationship detection dataset to push the frontier of human-interaction recognition. Finally the conclusion and the potential research directions are discussed in Chapter 8.

Chapter 2

Literature Review

In this part, I go through the related works in the literature. The topics of the thesis are 1) Designing energy-efficient neural networks and hashing methods for mobile devices, 2) visual recognition with deep neural networks. I will introduce each part in details.

2.1 Energy-efficient Neural Networks and Hashing

Deep convolutional neural networks (CNNs) have demonstrated record breaking results on a variety of computer vision tasks such as image classification [He et al., 2016a], semantic segmentation [Long et al., 2015] and object detection [Ren et al., 2015; Girshick et al., 2014]. Regardless of the availability of significantly improved training resources such as abundant annotated data, powerful computational platforms and diverse training frameworks, the promising results of deep CNNs are mainly attributed to the large number of learnable parameters, ranging from tens of millions to even hundreds of millions. However, this in turn lays heavy burdens on the memory and other computational resources. For instance, ResNet-152, a specific instance of the latest residual network architecture wining ImageNet classification challenge in 2015, has a model size of about 230MB and needs to perform about 11.3 billion FLOPs to classify a 224x224 image crop. Therefore, it is very challenging to deploy deep CNNs on the devices with limited computation and power budgets.

In another aspect, it becomes more necessary to cope with large-scale datasets with millions of images. Hashing methods construct a set of hash functions that map

the original features into binary codes, which enables fast nearest neighbor search by using look-up tables or Hamming distance based ranking. Moreover, compact binary codes are extremely efficient for large-scale data storage.

2.1.1 Energy Efficient Neural Networks

Several methods have been proposed to compress deep models and accelerate inference during testing. We can roughly summarize them into four main categories: quantizing parameters, low rank approximations, low-power network structure design and network pruning.

Limited numerical precision When deploying DNNs into hardware chips like FPGA, network quantization is a must process for efficient computing and storage. Several works have been proposed to quantize only parameters with high accuracy [Courbariaux et al., 2015; Zhu et al., 2017; Zhou et al., 2017]. Courbariaux *et al.*[Courbariaux et al., 2015] propose to constrain the weights to binary values (i.e., -1 or 1) to replace multiply-accumulate operations by simple accumulations. To keep a balance between the efficiency and the accuracy, ternary networks [Zhu et al., 2017] are proposed to keep the weights to 2bits while maintaining high accuracy. Zhou *et al.*[Zhou et al., 2017] presents incremental network quantization (INQ) to efficiently convert any pre-trained full-precision CNN model into low-precision whose weights are constrained to be either powers of two or zero.

Low-rank approximation Among existing works, some methods attempt to approximate low-rank filters in pre-trained networks [Kim et al., 2015; Zhang et al., 2016b]. Zhang *et al.*[Zhang et al., 2016b], reconstruction error of the nonlinear responses are minimized layer-wisely, with subject to the low-rank constraint to reduce the computational cost. Other seminal works attempt to restrict filters with low-rank constraints during training phrase [Novikov et al., 2015; Tai et al., 2015]. To better exploit the structure in kernels, it is also proposed to use low-rank tensor decomposition approaches [Denton et al., 2014; Novikov et al., 2015] to remove the redundancy

in convolutional kernels in pretrained networks.

Efficient architecture design The increasing demand for running highly energy efficient neural networks for hardware devices has motivated the network architecture design. GoogLeNet [Szegedy et al., 2015] and SqueezeNet [Iandola et al., 2016] propose to replace 3x3 convolutional filters with 1x1 size, which tremendously increase the depth of the network while decreasing the complexity a lot. ResNet [He et al., 2016a] and its variants [Zagoruyko and Komodakis, 2016; He et al., 2016b] utilize residual connections to relieve the gradient vanishing problem when training very deep networks. Recently, depthwise separable convolution employed in Xception [Chollet, 2016] and MobileNet [Howard et al., 2017] have been proved to be quite effective. Based on it, ShuffleNet [Zhang et al., 2017c] generalizes the group convolution and the depthwise separable convolution to get the state-of-the-art results.

Pruning and Sparsity Substantial effort have been made to reduce the storage of deep neural networks in order to save the bandwidth for dedicated hardware design. Han *et al.*[Han et al., 2015, 2016] introduce "deep compression", a three stage pipeline: pruning, trained quantization and Huffman coding to effectively reduce the memory requirement of CNNs with no loss of accuracy. Guo *et al.*[Guo et al., 2016] further incorporate connection slicing to avoid incorrect pruning. More works [Wen et al., 2016; Lebedev and Lempitsky, 2016; Liu et al., 2015] propose to employ structural sparsity for more energy-efficient compression.

2.1.2 Hashing

Hashing methods may be roughly categorized into data-dependent and data-independent schemes. Data-independent methods [Gionis et al., 1999; Kulis and Grauman, 2009; Jiang et al., 2015] focus on using random projections to construct random hash functions. The canonical example is the locality-sensitive hashing (LSH) [Gionis et al., 1999], which offers guarantees that metric similarity is preserved for sufficiently long codes based on random projections. Recent research focuses have been

shifted to data-dependent methods, which learn hash functions in a either unsupervised, semi-supervised, or supervised learning fashion. Unsupervised hashing methods [Carreira-Perpinan and Raziperchikolaei, 2015; Gong et al., 2013; Liu et al., 2011; Weiss et al., 2009, 2012; Shen et al., 2013] try to map the original features into hamming space while preserving similarity relations between the original features using unlabeled data. Supervised methods [Erin Liong et al., 2015; Shen et al., 2015; Kulis and Darrell, 2009; Liu et al., 2012; Li et al., 2013] use labelled training data for the similarity relations, aiming to preserve the "ground truth" similarity in the hash codes. Semi-supervised hashing methods incorporate ground-truth similarity information for the subset of the training data for which it is available, but also use unlabeled data.

Our proposed method belongs to the supervised hashing framework. Recently hashing using deep learning has shown great promise. The authors of [Zhao et al., 2015; Lai et al., 2015] learn hash bits such that multilevel semantic similarities are kept, taking raw pixels as input and training a deep CNN. This has the effect of simultaneously learning an image feature representation (in the early layers of the network) and the hash bits, which are obtained by thresholding the outputs of the last network layer, or *hash layer* at 0.5.

Note that these methods suffer from huge computation complexity introduced by the triplet ranking loss for hashing. In contrast, our proposed method is much more efficient in training, as shown in our experiments.

2.2 Deep Neural Networks for Visual Recognition

Convolutional neural networks (CNN) have been successfully applied in many visual recognition tasks, especially for image classification and object detection. In this section, we will first overview the backgrounds in general supervised image classification task and further in webly-supervised image classification. Moreover, we will then introduce the literature in classic object detection and further expand to high

level visual relationship detection task.

2.2.1 Supervised Image Classification

Deep convolutional neural networks have led to tremendous breakthroughs in image classification task. The improvement can due to advances in three directions: building more complex models, designing effective strategies against overfitting and solving the gradient vanishing problem. First, neural networks are becoming more capable of fitting training data by increasing their representation power. Several works propose to increase depth [Simonyan and Zisserman, 2015] or width [Zagoruyko and Komodakis, 2016] by stacking more layers or neurons, respectively. Some works instead propose to design complex network structures by using smaller strides [Zeiler and Fergus, 2014], new nonlinear activations [Maas et al., 2013; He et al., 2015], and sophisticated layer designs [He et al., 2014]. What's more, better generalization is achieved by effective regularization [Hinton et al., 2012] and various data augmentation strategies [Szegedy et al., 2015]. Furthermore, He *et al.* [He et al., 2016a] propose a residual architecture to solve the gradient vanishing problem in extremely deep neural networks for better convergence.

2.2.2 Webly-supervised Image Classification

Large-scale datasets have pushed the frontier of supervised image classification. However, annotating a massive dataset is expensive and time-consuming. So a webly-supervised learning strategy is extremely necessary in real world applications. Extensive works have been proposed to learn from web-scale data and noisy labels [Fergus et al., 2010; Schroff et al., 2011; Xu et al., 2015; Chen and Gupta, 2015; Divvala et al., 2014; Krause et al., 2016; Chen et al., 2013; Niu et al., 2015; Reed et al., 2014; Sukhbaatar and Fergus, 2015; Xiao et al., 2015; Mnih and Hinton, 2012]. In terms of learning from Web data, in [Chen et al., 2013; Chen and Gupta, 2015], Chen *et al.*propose to pre-train CNN on simple examples and adapt it to harder images by

leveraging the structure of data and categories in a two-step manner. In contrast, we propose a simply-yet-effective end-to-end learning framework without pre-training. To better dealing with noise, some approaches [Xiao et al., 2015; Sukhbaatar et al., 2014] propose to add an extra noise layer into the network which adapts the network outputs to match the noisy label distribution. On the other hand, some approaches attempt to remove or correct noisy labels [Brodley and Friedl, 2011; Miranda et al., 2009]. However, because of the difficulty of separating correctly labeled hard samples from mislabeled ones, such a strategy can result in removing too many (correct) instances. Moreover, several label noise-robust algorithms [Beigman and Klebanov, 2009; Manwani and Sastry, 2013] are proposed to make classifiers robust to label noise. However, noise-robust methods seem to be adequate only for simple cases of label noise that can be safely managed by regularization. In this thesis, we instead propose to suppress label noise by unified two strategies without any strong assumptions.

2.2.3 Image Detection

Image Detection is a basic block in many real world applications such as autonomous driving, face detection, pedestrian detection and so on. To detect an object, the original methods propose to take a classifier for that object and evaluate it at various locations and scales in a test image. For example, deformable parts models (DPM) [Felzenszwalb et al., 2010] use a sliding window approach where the classifier runs at each evenly spaced locations over the entire image. With the rapid development of deep learning, R-CNN [Girshick et al., 2014] use region proposal methods to first generate potential bounding boxes and then extract deep features over each box for classification. To accelerate the inference pipeline, Fast-RCNN [Girshick, 2015] propose to add a ROI pooling layer to max pooling the features inside any valid region of interest into a small feature map with a fixed spatial extent. What's more, Ren *et al.* [Ren et al., 2015] further propose to end-to-end train a Region Proposal Network

(RPN) to generate high-quality region proposals, which are used by Fast R-CNN for detection. Recent works like [Redmon et al., 2016] unify the separate components of object detection into a single neural network for better speed and performance.

2.2.4 Visual Relationship Detection

However, object detection focuses on detecting individual objects such as woman, toothbrush, and child while they don't consider the semantic relationships between the detected objects. Understanding visual scenes is one of the primal goals of computer vision. For high-level understanding of the scene, the fundamental element is to model visual relationships, the mutual correlations of the detected objects in the scene. Visual relationships are not a new concept. It has been investigated by numerous studies in the last decade. In the early days, most works target specific types of phrases [Choi et al., 2013; Desai and Ramanan, 2012] or use visual phrases to improve other tasks [Sadeghi and Farhadi, 2011; Kumar and Koller, 2010; Russell et al., 2006]. For example, Sadeghi *et al.*has proved the phrase, as a whole, can facilitate object recognition because of its special visual appearance [Sadeghi and Farhadi, 2011]. Desai *et al.*use the phrase that describes the interaction between a person and objects to facilitate actions, pose and object detection [Desai and Ramanan, 2012]. Recently, researchers pay more attention to general visual relationship detection [Li et al., 2017a; Xu et al., 2017; Plummer et al., 2016; Zhang et al., 2017b; Zhuang et al., 2017b]. Lu, *et al.*first formalize the visual relationship detection as a task and propose the state-of-art method by leveraging the language prior to model the correlation between subject/object and predicate [Lu et al., 2016]. Li *et al.*use the message passing structure among subject, object and predicate branches to model their dependencies [Li et al., 2017a]. Xu *et al.*built up a fully-connected graph to iteratively pass messages along the scene graph [Xu et al., 2017]. Liang *et al.*applied the reinforcement learning method to the relationship and attribute detection [Liang et al., 2017b].

Chapter 3

Towards Effective Low-bitwidth Convolutional Neural Networks

Statement of Authorship

Title of Paper	Towards efficient low-bitwidth convolutional neural networks
Publication Status	☐ Published ☐ Accepted for Publication ☒ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Under peer review

Principal Author

Name of Principal Author (Candidate)	Bohan Zhuang
Contribution to the Paper	Wrote the paper and completed the experiments.
Overall percentage (%)	90%
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.
Signature	Date 7/12/2017

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Chunhua Shen
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/2017

Name of Co-Author	Mingkui Tan
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/2017

Please cut and paste additional co-author panels here as required.

Name of Co-Author	Lingqiao Liu		
Contribution to the Paper	Helped design the method and modified the paper.		
Signature		Date	7/12/2017

Name of Co-Author	Ian Reid		
Contribution to the Paper	Helped design the method and modified the paper.		
Signature		Date	7/12/17

3.1 Overview

This chapter tackles the problem of training a deep convolutional neural network with both low-precision weights and low-bitwidth activations. Optimizing a low-precision network is very challenging since the training process can easily get trapped in a poor local minima, which results in substantial accuracy loss. To mitigate this problem, we propose three simple-yet-effective approaches to improve the network training. First, we propose to use a two-stage optimization strategy to progressively find good local minima. Specifically, we propose to first optimize a net with quantized weights and then quantized activations. This is in contrast to the traditional methods which optimize them simultaneously. Second, following a similar spirit of the first method, we propose another progressive optimization approach which progressively decreases the bit-width from high-precision to low-precision during the course of training. Third, we adopt a novel learning scheme to jointly train a full-precision model alongside the low-precision one. By doing so, the full-precision model provides hints to guide the low-precision model training. Extensive experiments on various datasets (i.e., , CIFAR-100 and ImageNet) show the effectiveness of the proposed methods. To highlight, using our methods to train a 4-bit precision network leads to no performance decrease in comparison with its full-precision counterpart with standard network architectures (i.e., , AlexNet and ResNet-50).

3.2 Introduction

The state-of-the-art deep neural networks [Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016a] usually involve millions of parameters and need billions of FLOPs during computation. Those memory and computational cost can be unaffordable for mobile hardware device or especially implementing deep neural networks on chips. To improve the computational and memory efficiency, various solutions have been proposed, including pruning network weights [Han et al., 2015,

2016], low rank approximation of weights [Kim et al., 2015; Zhang et al., 2016b], and training a low-bit-precision network [Zhou et al., 2017; Courbariaux et al., 2015; Zhu et al., 2017; Zhou et al., 2016]. In this work, we follow the idea of training a low-precision network and our focus is to improve the training process of such a network. Note that in the literature, many works adopt this idea but only attempt to quantize the weights of a network while keeping the activations to 32-bit floating point [Zhou et al., 2017; Courbariaux et al., 2015; Zhu et al., 2017]. Although this treatment leads to lower performance decrease comparing to its full-precision counterpart, it still needs substantial amount of computational resource requirement to handle the full-precision activations. Thus, our work targets the problem of training network with *both low-bit quantized weights and activations.*

The solutions proposed in this chapter contain three components. They can be applied independently or jointly. The first method is to adopt a two-stage training process. At the first stage, only the weights of a network is quantized. After obtaining a sufficiently good solution of the first stage, the activation of the network is further required to be in low-precision and the network will be trained again. Essentially, this progressive approach first solves a related sub-problem, i.e., training a network with only low-bit weights and the solution of the sub-problem provides a good initial point for training our target problem. Following the similar idea, we propose our second method by performing progressive training on the bit-width aspect of the network. Specifically, we incrementally train a serial of networks with the quantization bit-width (precision) gradually decreased from full-precision to the target precision. The third method is inspired by the recent progress of mutual learning [Zhang et al., 2017d] and information distillation [Romero et al., 2015; Hinton et al., 2015; Parisotto et al., 2016; Zagoruyko and Komodakis, 2017; Ba and Caruana, 2014]. The basic idea of those works is to train a target network alongside another guidance network. For example, The works in [Romero et al., 2015; Hinton et al., 2015; Parisotto et al., 2016; Zagoruyko and Komodakis, 2017; Ba and Caruana, 2014] propose to train a small

student network to mimic the deeper or wider teacher network. They add an additional regularizer by minimizing the difference between student's and teacher's posterior probabilities [Hinton et al., 2015] or intermediate feature representations [Ba and Caruana, 2014; Romero et al., 2015]. It is observed that by using the guidance of the teacher model, better performance can be obtained with the student model than directly training the student model on the target problem. Motivated by these observations, we propose to train a full-precision network alongside the target low-precision network. Also, in contrast to standard knowledge distillation methods, we do not require to pre-train the guidance model. Rather, we allow the two models to be trained jointly from scratch since we discover that this treatment enables the two nets adjust better to each other.

Compared to several existing works that achieve good performance when quantizing both weights and activations [Wu et al., 2016a; Zhou et al., 2016; Hubara et al., 2016; Rastegari et al., 2016], our methods is more considerably scalable to the deeper neural networks [He et al., 2016a,b]. For example, some methods adopt a layer-wise training procedure [Wu et al., 2016a], thus their training cost will be significantly increased if the number of layers becomes larger. In contrast, the proposed method does not have this issue and we have experimentally demonstrated that our method is effective with various depth of networks (i.e., , AlexNet, ResNet-50).

3.3 Related work

Several methods have been proposed to compress deep models and accelerate inference during testing. We can roughly summarize them into four main categories: limited numerial percision, low-rank approximation, efficient architecture design and network pruning.

Limited numerical precision When deploying DNNs into hardware chips like FPGA, network quantization is a must process for efficient computing and storage. Several works have been proposed to quantize only parameters with high accu-

racy [Courbariaux et al., 2015; Zhu et al., 2017; Zhou et al., 2017]. Courbariaux *et al.*[Courbariaux et al., 2015] propose to constrain the weights to binary values (i.e., , -1 or 1) to replace multiply-accumulate operations by simple accumulations. To keep a balance between the efficiency and the accuracy, ternary networks [Zhu et al., 2017] are proposed to keep the weights to 2-bit while maintaining high accuracy. Zhou *et al.*[Zhou et al., 2017] present incremental network quantization (INQ) to efficiently convert any pre-trained full-precision CNN model into low-precision whose weights are constrained to be either powers of two or zero. Different from these methods, a mutual knowledge transfer strategy is proposed to jointly optimize the full-precision model and its low-precision counterpart for high accuracy. What's more, we propose to use a progressive optimization approach to quantize both weights and activations for better performance.

Low-rank approximation Among existing works, some methods attempt to approximate low-rank filters in pre-trained networks [Kim et al., 2015; Zhang et al., 2016b]. In [Zhang et al., 2016b], reconstruction error of the nonlinear responses are minimized layer-wisely, with subject to the low-rank constraint to reduce the computational cost. Other seminal works attempt to restrict filters with low-rank constraints during training phrase [Novikov et al., 2015; Tai et al., 2015]. To better exploit the structure in kernels, it is also proposed to use low-rank tensor decomposition approaches [Denton et al., 2014; Novikov et al., 2015] to remove the redundancy in convolutional kernels in pretrained networks.

Efficient architecture design The increasing demand for running highly energy efficient neural networks for hardware devices have motivated the network architecture design. GoogLeNet [Szegedy et al., 2015] and SqueezeNet [Iandola et al., 2016] propose to replace 3x3 convolutional filters with 1x1 size, which tremendously increase the depth of the network while decreasing the complexity a lot. ResNet [He et al., 2016a] and its variants [Zagoruyko and Komodakis, 2016; He et al., 2016b] utilize residual connections to relieve the gradient vanishing problem when training

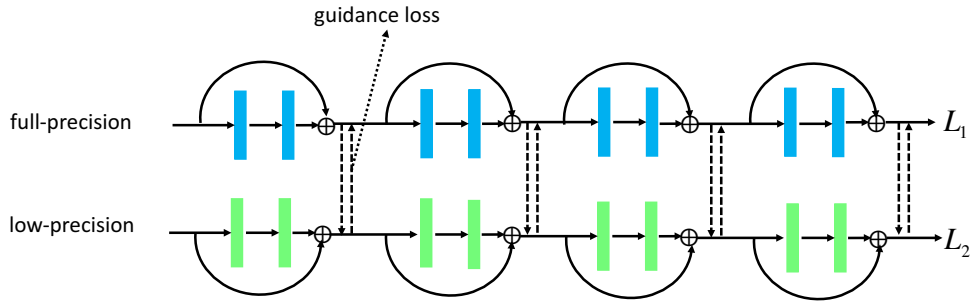


Figure 3.1: Demonstration of the guided training strategy. We use the residual network structure for illustration.

very deep networks. Recently, depthwise separable convolution employed in Xception [Chollet, 2016] and MobileNet [Howard et al., 2017] have been proved to be quite effective. Based on it, ShuffleNet [Zhang et al., 2017c] generalizes the group convolution and the depthwise separable convolution to get the state-of-the-art results.

Pruning and sparsity Substantial effort have been made to reduce the storage of deep neural networks in order to save the bandwidth for dedicated hardware design. Han *et al.*[Han et al., 2015, 2016] introduce "deep compression", a three stage pipeline: pruning, trained quantization and Huffman coding to effectively reduce the memory requirement of CNNs with no loss of accuracy. Guo *et al.*[Guo et al., 2016] further incorporate connection slicing to avoid incorrect pruning. More works [Wen et al., 2016; Lebedev and Lempitsky, 2016; Liu et al., 2015] propose to employ structural sparsity for more energy-efficient compression.

3.4 Methods

In this section, we will first revisit the quantization function in the neural network and the way to train it. Then we will elaborate our three methods in the subsequent sections.

3.4.1 Quantization function revisited

A common practise in training a neural network with low-precision weights and activations is to introduce a quantization function. Considering the general case of k-bit quantization as in [Zhou et al., 2016], we define the quantization function $Q(\cdot)$ to be

$$z_q = Q(z_r) = \frac{1}{2^k - 1} round((2^k - 1)z_r) \tag{3.1}$$

where $z_r \in [0, 1]$ denotes the full-precision value and $z_q \in [0, 1]$ denotes the quantized value. With this quantization function, we can define the weight quantization process and the activation quantization process as follows:

Quantization on weights:

$$w_q = Q(\frac{\tanh(w)}{2\max(|\tanh(w)|)} + \frac{1}{2}). \tag{3.2}$$

In other words, we first use $\frac{\tanh(w)}{2\max(|\tanh(w)|)} + \frac{1}{2}$ to obtain a normalized version of w and then perform the quantization, where $\tanh(\cdot)$ is adopted to reduce the impact of large values.

Quantization on activations:

Same as [Zhou et al., 2016], we first use a clip function $f(x) = clip(x, 0, 1)$ to bound the activations to $[0, 1]$. After that, we conduct quantize the activation by applying the quantization function $Q(\cdot)$ on $f(x)$.

$$x_q = Q(f(x)). \tag{3.3}$$

Back-propagation with quantization function: In general, the quantization function is non-differentiable and thus it is impossible to directly apply the back-propagation to train the network. To overcome this issue, we adopt the straight-through estimator [Zhou et al., 2016; Hubara et al., 2016; Bengio et al., 2013] to approximate the gradients calculation. Formally, we approximate the partial gradient $\frac{\partial z_q}{\partial z_r}$ with an identity

mapping, namely $\frac{\partial z_q}{\partial z_r} \approx 1$. Accordingly, $\frac{\partial l}{\partial z_r}$ can be approximated by

$$\frac{\partial l}{\partial z_r} = \frac{\partial l}{\partial z_q}\frac{\partial z_q}{\partial z_r} \approx \frac{\partial l}{\partial z_q}. \tag{3.4}$$

3.4.2 Two-stage optimization

With the straight-through estimator, it is possible to directly optimize the low-precision network. However, the gradient approximation of the quantization function inevitably introduces noisy signal for updating network parameters. Strictly speaking, the approximated gradient may not be the right updating direction. Thus, the training process will be more likely to get trapped at a poor local minima than training a full precision model. Applying the quantization function to both weights and activations further worsens the situation.

To reduce the difficulty of training, we devise a two-stage optimization procedure: at the first stage, we only quanitze the weights of the network while setting the activations to be full precision. After the converge (or after certain number of iterations) of this model, we further apply the quantization function on the activations as well and retrain the network. Essentially, the first stage of this method is a related subproblem of the target one. Compared to the target problem, it is easier to optimize since it only introduces quantization function on weights. Thus, we are more likely to arrive at a good solution for this sub-problem. Then, using it to initialize the target problem may help the network avoid poor local minima which will be encountered if we train the network from scratch. Let M_{low}^K be the high-precision model with K-bit. We propose to learn a low-precision model M_{low}^k in a two-stage manner with M_{low}^K serving as the initial point, where $k < K$. The detailed algorithm is shown in Algorithm 1.

Algorithm 1: Two-stage optimization for k-bit quantization

Input: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$; A K-bit precision model M_{low}^{K}.
Output: A low-precision deep model M_{low}^{k} with weights $\mathbf{W}_{low}$ and activations being quantized into k-bit.

Stage 1: Quantize $\mathbf{W}_{low}$:
for epoch $= 1, ..., L$ **do**
  for $t = 1, ...T$ **do**
    Randomly sample a mini-batch data;
    Quantize the weights $\mathbf{W}_{low}$ into k-bit by calling some quantization methods with K-bit activations;
Stage 2: Quantize activations:
Initialize $\mathbf{W}_{low}$ using the converged k-bit weights from **Stage 1** as the starting point;
for epoch $= 1, ..., L$ **do**
  for $t = 1, ...T$ **do**
    Randomly sample a mini-batch data;
    Quantize the activations into k-bit by calling some quantization methods while keeping the weights to k-bit;

3.4.3 Progressive quantization

The aforementioned two-stage optimization approach suggests the benefits of using a related easy optimized problem to find a good initialization. However, separating the quantization of weights and activations is not the only solution to implement the above idea. In this chapter, we also propose another solution which progressively lower the bitwidth of the quantization during the course of network training. Specifically, we progressively conduct the quantization from higher precisions to lower precisions (e.g., , 32-bit $\rightarrow$ 8-bit $\rightarrow$ 4-bit $\rightarrow$ 2-bit).[1]. The model of higher precision will be used the the starting point of the relatively lower precision, in analogy with annealing.

Let $\{b_1, ..., b_n\}$ be a sequence precisions, where $b_n < b_{n-1}, ..., b_2 < b_1$, b_n is the target precision and b_1 is set to 32 by default. The whole progressive optimization procedure is summarized in as Algorithm 2. Let M_{low}^{k} be the low-precision model

[1]We notice in practice that there is virtually no loss of accuracy in skipping directly from 32 to 8 bits without first passing through intermediate precisions.

with k-bit and M_{full} be the full precision model. In each step, we propose to learn M_{low}^{k}, with the solution in the $(i-1)$-th step, denoted by M_{low}^{K}, serving as the initial point, where $k < K$.

Algorithm 2: Progressive quantization for accurate CNNs with low-precision weights and activations

Input: Training data $\{(\mathbf{x}_j, y_j)\}_{j=1}^{N}$; A pre-trained 32-bit full-precision model M_{full} as baseline; the precision sequence $\{b_1, ..., b_n\}$ where $b_n < b_{n-1}, ..., b_2 < b_1 = 32$.

Output: A low-precision deep model $M_{low}^{b_n}$.

```
Let M_low^{b_1} = M_full, where b_1 = 32;
for i = 2, ...n do
    Let k = b_i and K = b_{i-1};
    Obtain M_low^k by calling some quantization methods with M_low^K being the input;
```

3.4.4 Guided training with a full-precision network

The third method proposed in this chapter is inspired by the success of using information distillation [Romero et al., 2015; Hinton et al., 2015; Parisotto et al., 2016; Zagoruyko and Komodakis, 2017; Ba and Caruana, 2014] to train a relatively shallow network. Specifically, these methods usually use a teacher model (usually a pretrained deeper network) to provide guided signal for the shallower network. Following this spirit, we propose to train the low-precision network alongside another guidance network. Unlike the work in [Romero et al., 2015; Hinton et al., 2015; Parisotto et al., 2016; Zagoruyko and Komodakis, 2017; Ba and Caruana, 2014], the guidance network shares the same architecture as the target network but is pretrained with full-precision weights and activations.

However, a pre-trained model may not be necessarily optimal or may not be suitable for quantization. As a result, directly using a fixed pretrained model to guide the target network may not produce the best guidance signals. To mitigate this problem, we do not fix the parameters of a pretrained full precision network as in the previous work [Zhang et al., 2017d].

By using the guidance training strategy, we assume that there exist some full-precision models with good generalization performance, and an accurate low-precision model can be obtained by directly performing the quantization on those full-precision models. In this sense, the feature maps of the learned low-precision model should be close to that obtained by directly doing quantization on the full-precision model. To achieve this, essentially, in our learning scheme, we can jointly train the full-precision and low-precision models. This allows these two models adapt to each other. We even find by doing so the performance of the full-precision model can be slightly improved in some cases.

Formally, let $\mathbf{W}_{full}$ and $\mathbf{W}_{low}$ be the weights of the full-precision model and low-precision model, respectively. Let $\mu(\mathbf{x}; \mathbf{W}_{full})$ and $\nu(\mathbf{x}; \mathbf{W}_{low})$ be the nested feature maps (e.g., activations) of the full-precision model and low-precision model, respectively. To create the guidance signal, we may require that the nested feature maps from the two models should be similar. However, $\mu(\mathbf{x}; \mathbf{W}_{full})$ and $\nu(\mathbf{x}; \mathbf{W}_{low})$ is usually not directly comparable since one is full precision and the other is low-precision.

To link these two models, we can directly quantize the weights and activations of the full-precision model by equations (3.2) and (3.3). For simplicity, we denote the quantized feature maps by $Q(\mu(\mathbf{x}; \mathbf{W}_{full}))$. Thus, $Q(\mu(\mathbf{x}; \mathbf{W}_{full}))$ and $\nu(\mathbf{x}; \mathbf{W}_{low})$ will become comparable. Then we can define the guidance loss as:

$$R(\mathbf{W}_{full}, \mathbf{W}_{low}) = \frac{1}{2} \parallel Q(\mu(\mathbf{x}; \mathbf{W}_{full})) - \nu(\mathbf{x}; \mathbf{W}_{low}) \parallel^2, \tag{3.5}$$

where $\parallel \cdot \parallel$ denotes some proper norms.

Let L_{θ_1} and L_{θ_2} be the cross-entropy classification losses for the full-precision and low-precision model, respectively. The guidance loss will be added to L_{θ_1} and L_{θ_2}, respectively, resulting in two new objectives for the two networks, namely

$$L_1(\mathbf{W}_{full}) = L_{\theta_1} + \lambda R(\mathbf{W}_{full}, \mathbf{W}_{low}). \tag{3.6}$$

and

$$L_2(\mathbf{W}_{low}) = L_{\theta_2} + \lambda R(\mathbf{W}_{full}, \mathbf{W}_{low}). \tag{3.7}$$

where λ is a balancing parameter. Here, the guidance loss R can be considered as some regularization on L_{θ_1} and L_{θ_2}.

In the learning procedure, both $\mathbf{W}_{full}$ and $\mathbf{W}_{low}$ will be updated by minimizing $L_1(\mathbf{W}_{full})$ and $L_2(\mathbf{W}_{low})$ separately, using a mini-batch stochastic gradient descent method. The detailed algorithm is shown in Algorithm 3. A high-bit precision model M_{low}^K is used as an initialization of M_{low}^k, where $K > k$. Specifically, for the full-precision model, we have $K = 32$. Relying on M_{full}, the weights and activations of M_{low}^k can be initialized by equations (3.2) and (3.3), respectively.

Note that the training process of the two networks are different. When updating $\mathbf{W}_{low}$ by minimizing $L_2(\mathbf{W}_{low})$, we use full-precision model as the initialization and apply the forward-backward propagation rule in Section 3.4.1 to fine-tune the model. When updating $\mathbf{W}_{full}$ by minimizing $L_1(\mathbf{W}_{full})$, we use conventional forward-backward propagation to fine-tune the model.

Algorithm 3: Guided training with a full-precision network for k-bit quantization

Input: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$; A pre-trained 32-bit full-precision model M_{full}; A k-bit precision model M_{low}^k.

Output: A low-precision deep model M_{low}^k with weights and activations being quantized into k bits.

```
Initialize M_low^k based on M_full;
for epoch = 1, ..., L do
    for t = 1, ...T do
        Randomly sample a mini-batch data;
        Quantize the weights W_low and activations into k-bit by minimizing
        L_2(W_low);
        Update M_full by minimizing L_1(W_full);
```

3.4.5 Remark on the proposed methods

The proposed three approaches tackle the difficulty in training a low-precision model with different strategies. They can be applied independently. However, it is also possible to combine them together. For example, we can apply the progressive quantization to any of the steps in the two-stage approach; we can also apply the guided training to any sub-step in the progressive training. Detailed analysis on possible combinations will be experimentally evaluated in the experiment section.

3.4.6 Implementation details

In all the three methods, we quantize the weights and activations of all layers except that the input data are kept to 8-bit. Furthermore, to promote convergence, we propose to add a scalar layer after the last fully-connected layer before feeding the low-bit activations into the softmax function for classification. The scalar layer has only one trainable small scalar parameter and is initialized to 0.01 in our approach.

During training, we randomly crop 224x224 patches from an image or its horizontal flip, with the per-pixel mean subtracted. We don't use any further data augumentation in our implementation. We adopt batch normalization (BN) [Ioffe and Szegedy, 2015] after each convolution before activation. For pretraining the full-precision baseline model, we use Nesterov SGD and batch size is set to 256. The learning rate starts from 0.01 and is divided by 10 every 30 epochs. We use a weight decay 0.0001 and a momentum 0.9. For weights and activations quantization, the initial learning rate is set to 0.001 and is divided by 10 every 10 epochs. We use a simple single-crop testing for standard evaluation. Following [Zagoruyko and Komodakis, 2017], for ResNet-50, we add only two guidance losses in the 2 last groups of residual blocks. And for AlexNet, we add two guidance losses in the last two fully-connected layers.

3.5 Experiment

To investigate the performance of the proposed methods, we conduct experiments on Cifar100 and ImageNet datasets. Two representative networks, different precisions AlexNet and ResNet-50 are evaluated with top-1 and top-5 accuracy reported. We use a variant of AlexNet structure [Krizhevsky et al., 2012] by removing dropout layers and add batch normalization after each convolutional layer and fully-connected layer. This structure is widely used in previous works [Zhou et al., 2016; Zhu et al., 2017]. We analyze the effect of the guided training approach, two-stage optimization and the progressive quantization in details in the ablation study. Seven methods are implemented and compared:

1. "**Baseline**": We implement the baseline model based on DoReFa-Net as described in Section 3.4.1.

2. "**TS**": We apply the two-stage optimization strategy described in Sec. 3.4.2 and Algorithm 1 to quantize the weights and activations. We denote the first stage as **Stage1** and the second stage as **Stage2**.

3. "**PQ**": We apply the progressive quantization strategy described in Sec. 3.4.3 and Algorithm 2 to continuously quantize weights and activations simultaneously from high-precision (i.e., , 32-bit) to low-precision.

4. "**Guided**": We implement the guided training approach as described in Sec. 3.4.4 and Algorithm 3 to independently investigate its effect on the final performance.

5. "**PQ+TS**": We further combine **PQ** and **TS** together to see whether their combination can improve the performance.

6. "**PQ+TS+Guided**": This implements the full model by combining **PQ**, **TS** and **Guided** modules together.

7. "**PQ+TS+Guided****": Based on **PQ+TS+Guided**, we use full-precision weights for the first convolutional layer and the last fully-connected layer following the setting of [Zhu et al., 2017; Zhou et al., 2016] to investigate its sensitivity to the proposed method.

3.5.1 Evaluation on ImageNet

We further train and evaluate our model on ILSVRC2012 [Russakovsky et al., 2015b], which includes over 1.2 million images and 50 thousand validation images. We report 4-bit and 2-bit precision accuracy for both AlexNet and ResNet-50. The sequence of bit-width precisions are set as $\{32, 8, 4, 2\}$. The results of INQ [Zhou et al., 2017] are directly cited from the original paper. We did not use the sophisticated image augmentation and more details can be found in Sec. 6.3.4. We compare our model to the 32-bit full-precision model, INQ, DoReFa-Net and the baseline approach described in Sec. 3.4.1. For INQ, only the weights are quantized. For DoReFa-Net, the first convolutional layer uses the full-precision weights and the last fully-connected layer use both full-precision weights and activations.

Results on AlexNet: The results for AlexNet are listed in Table 3.1. Compared to competing approaches, we achieve steadily improvement for 4-bit and 2-bit settings. This can be attributed to the effective progressive optimization and the knowledge from the full-precision model for assisting the optimization process. Furthermore, our 4-bit full model even outperforms the full-precision reference by 0.7% on top-1 accuracy. This may be due to the fact that on this data, we may not need a model as complex as the full-precision one. However, when the expected bit-width decrease to 2-bit, we observe obvious performance drop compared to the 32-bit model while our low-bit model still brings 2.8% top-1 accuracy increase compared to the *Baseline* method.

Results on ResNet-50: The results for ResNet-50 are listed in Table 3.2. For the full-precision model, we implement it using Pytorch following the re-implementation

Accuracy	Full precision	5-bit (INQ)	4-bit (DoReFa-Net)	4-bit (Baseline)	4-bit (PQ+TS+Guided)	2-bit (DoReFa-Net)	2-bit (Baseline)	2-bit (PQ+TS+Guided)
Top1	57.2%	57.4%	56.2%	56.8%	**58.0%**	48.3%	48.8%	**51.6%**
Top5	80.3%	80.6%	79.4%	80.0%	**81.1%**	71.6%	72.2%	**76.2%**

Table 3.1: Top1 and Top5 validation accuracy of AlexNet on ImageNet.

Accuracy	Full precision	5-bit (INQ)	4-bit (DoReFa-Net)	4-bit (Baseline)	4-bit (PQ+TS+Guided)	2-bit (DoReFa-Net)	2-bit (Baseline)	2-bit (PQ+TS+Guided)
Top1	75.6%	74.8%	74.5%	75.1%	**75.7%**	67.3%	67.7%	**70.0%**
Top5	92.2%	91.7%	91.5%	91.9%	**92.0%**	84.3%	84.7%	**87.5%**

Table 3.2: Top1 and Top5 validation accuracy of ResNet-50 on ImageNet.

provided by Facebook[2]. Comparatively, we find that the performance are approximately consistent with the results of AlexNet. Similarly, we observe that our 4-bit full model is comparable with the full-precision reference with no loss of accuracy. When decreasing the precision to 2-bit, we achieve promising improvement over the competing *Baseline* even though there's still an accuracy gap between the full-precision model. Similar to the AlexNet on ImageNet dataset, we find our 2-bit full model improves more comparing with the 4-bit case. This phenomenon shows that when the model becomes more difficult to optimize, the proposed approach turns out to be more effective in dealing with the optimization difficulty. To better understand our model, we also draw the process of training for 2-bit ResNet-50 in Figure 3.3 and more analysis can be referred in Sec. 3.5.3.

3.5.2 Evaluation on Cifar100

Cifar100 is an image classification benchmark containing images of size 32x32 in a training set of 50,000 and a test set of 10,000. We use the AlexNet for our experiment. The quantitative results are reported in Table 3.3. From the table, we can observe that the proposed approach steadily outperforms the competing method DoReFa-Net. Interestingly, the accuracy of our 4-bit full model also surpasses its full precision model. We speculate that this is due to 4-bit weights and activations providing the right model capacity and preventing overfitting for the networks.

[2] `https://github.com/facebook/fb.resnet.torch`

Accuracy	Full precision	4-bit (DoReFa-Net)	4-bit (Baseline)	4-bit (PQ+TS+Guided)	2-bit (DoReFa-Net)	2-bit (Baseline)	2-bit (PQ+TS+Guided)
Top1	65.4%	64.9%	65.0%	**65.8%**	63.4%	63.9%	**64.6%**
Top5	88.3%	88.5%	88.5%	**88.6%**	87.5%	87.6%	**87.8%**

Table 3.3: Top1 and Top5 validation accuracy of AlexNet on Cifar100.

Method	top-1	top-5
4-bit (TS)	57.7%	81.0%
4-bit (PQ)	57.5%	80.8%
4-bit (PQ+TS)	57.8%	80.8%
4-bit (Guided)	57.3%	80.4%
4-bit (PQ+TS+Guided)	58.0%	81.1%
4-bit (PQ+TS+Guided**)	**58.1%**	**81.2%**
2-bit (TS)	50.7%	74.9%
2-bit (PQ)	50.3%	74.8%
2-bit (PQ+TS)	50.9%	74.9%
2-bit (Guided)	50.0%	74.1%
2-bit (PQ+TS+Guided)	51.6%	76.2%
2-bit (PQ+TS+Guided**)	**52.5%**	**77.3%**

Table 3.4: Evaluation of different components of the proposed method on the validation accuracy with AlexNet on ImageNet.

3.5.3 Ablation study

In this section, we analyze the effects of different components of the proposed model.

Learning from scratch vs. Fine-tuning: To analyze the effect, we perform comparative experiments on Cifar100 with AlexNet using learning from scratch and fine-tuning strategies. The results are shown in Figure 3.2, respectively. For convenience of exposition, this comparison study is performed based on method *TS*. First, we observe that the overall accuracy of fine-tuning from full-precision model is higher than that of learning from scratch. This indicates that the initial point for training low-bitwidth model is crutial for obtaining good accuracy. In addition, the gap between the *Baseline* and *TS* is obvious (i.e., , 2.7 % in our experiment) with learning from scratch. This justifies that the two-stage optimization strategy can effectively help the model converge to a better local minimum.

The effect of quantizing all layers: This set of experiments is performed to analyze the influence for quantizing the first convolutional layer and the last fully-connected layer. Several previous works [Zhu et al., 2017] argue to keep these two layers precision as 32-bit floating points to decrease accuracy loss. By comparing the results

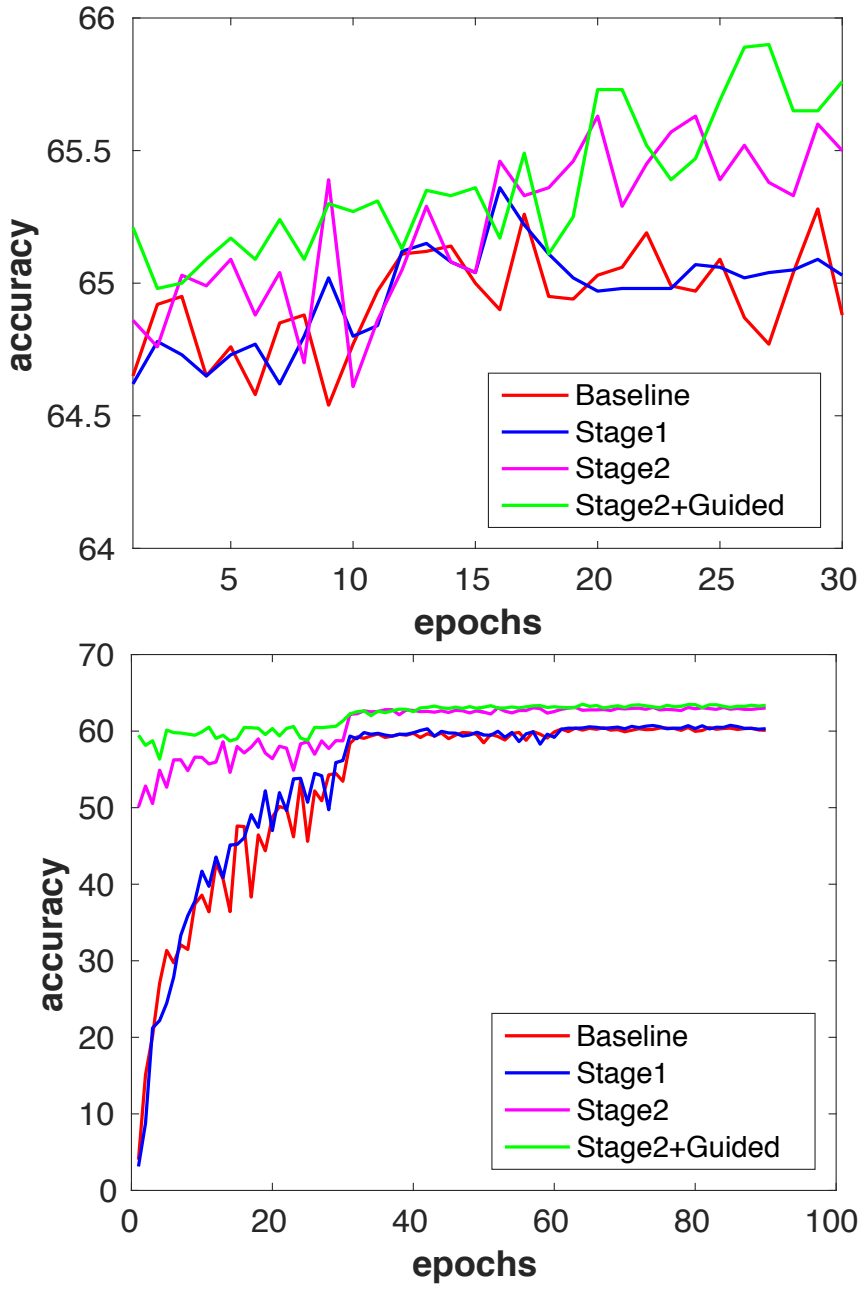


Figure 3.2: Validation accuracy of 4-bit AlexNet on Cifar100 using (a): the fine-tuning strategy; (b): learning from scratch strategy. *Stage2+Guided* means we combine the methods *Stage2* and *Guided* together during optimization to investigate the effect of the guided training on the final performance.

Method	top-1	top-5
4-bit (TS)	75.3%	91.9%
4-bit (PQ)	75.4%	91.8%
4-bit (PQ+TS)	75.5%	92.0%
4-bit (Guided)	75.3 %	91.7%
4-bit (PQ+TS+Guided)	75.7%	92.0%
4-bit (PQ+TS+Guided**)	**75.9%**	**92.4%**
2-bit (TS)	69.2%	87.0%
2-bit (PQ)	68.8%	86.9%
2-bit (PQ+TS)	69.4%	87.0%
2-bit (Guided)	69.0%	86.8%
2-bit (PQ+TS+Guided)	70.0%	87.5%
2-bit (PQ+TS+Guided**)	**70.8%**	**88.3%**

Table 3.5: Evaluation of different components of the proposed method on the validation accuracy with ResNet-50 on ImageNet.

of *PQ+TS+Guided*** and *PQ+TS+Guided* in Table 3.4 and Table 3.5, we notice that the accuracy gap between the two settings is not large, which indicates that our model is not sensitive to the precision of these two layers. It can be attributed to two facts. On one hand, fine-tuning from 32-bit precision can drastically decrease the difficulty for optimization. On the other hand, the progressive optimization approach as well as the guided training strategy further ease the instability during training.

The effect of the two-stage optimization strategy: We further analyze the effect of each stage in the *TS* approach in Figure 3.2 and Figure 3.3. We take the 2-bitwidth ResNet-50 on ImageNet as an example. In Figure 3.3, *Stage1* has the minimal loss of accuracy. As for the *Stage2*, although it incurs apparent accuracy decrease in comparison with that of the *Stage1*, its accuracy is consistently better than the results of *Baseline* in every epoch. This illustrates that progressively seeking for the local minimum point is crutial for final better convergence. We also conduct additional experiments on Cifar100 with 4-bit AlexNet. Interestingly, taking the model of *Stage1* as the initial point, the results of *Stage2* even have relative increase using two different training strategies as mentioned above. This can be interpreted by that further quantizing the activations impose more regularization on the model to overcome overfitting. Overall, the two-step optimization strategy still performs steadily better than the Baseline method which proves the effectiveness of this simple mechanism.

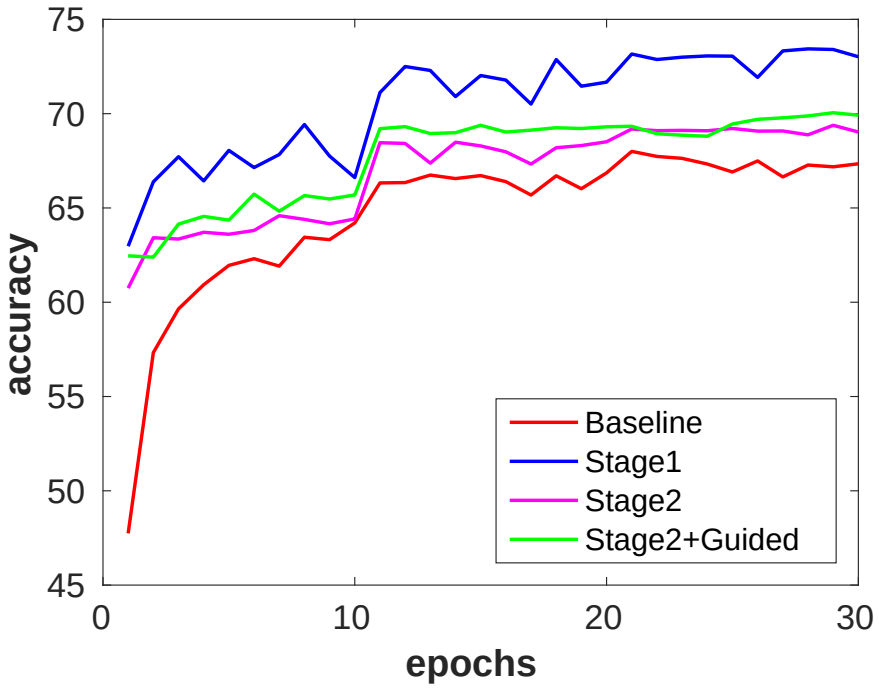


Figure 3.3: Validation accuracy of 2-bit ResNet-50 on ImageNet. *Stage2+Guided* means we combine the methods *Stage2* and *Guided* together during training.

The effect of the progressive quantization strategy: What's more, we also separately explore the progressive quantization (i.e., , *PQ*) effect on the final performance. In this experiment, we apply AlexNet on the ImageNet dataset. We continuously quantize both weights and activations simultaneously from 32-bit→8-bit→4-bit→2-bit and explictly illustrate the accuracy change process for each precision in Figure 3.4. The quantitative results are also reported in Table 3.4 and Table 3.5. From the figure we can find that for the 8-bit and 4-bit, the low-bit model has no accuracy loss with respect to the full precision model. However, when quantizing from 4-bit to 2-bit, we can observe significant accuracy drop. Despite this, we still observe 1.5% relative improvement by comparing the top-1 accuracy over the 2-bit baseline, which proves the effectiveness of the proposed strategy. It is worth noticing that the accuracy curves become more unstable when quantizing to lower bit. This phenomenon is reasonable since the precision becomes lower, the value will change more frequently during training.

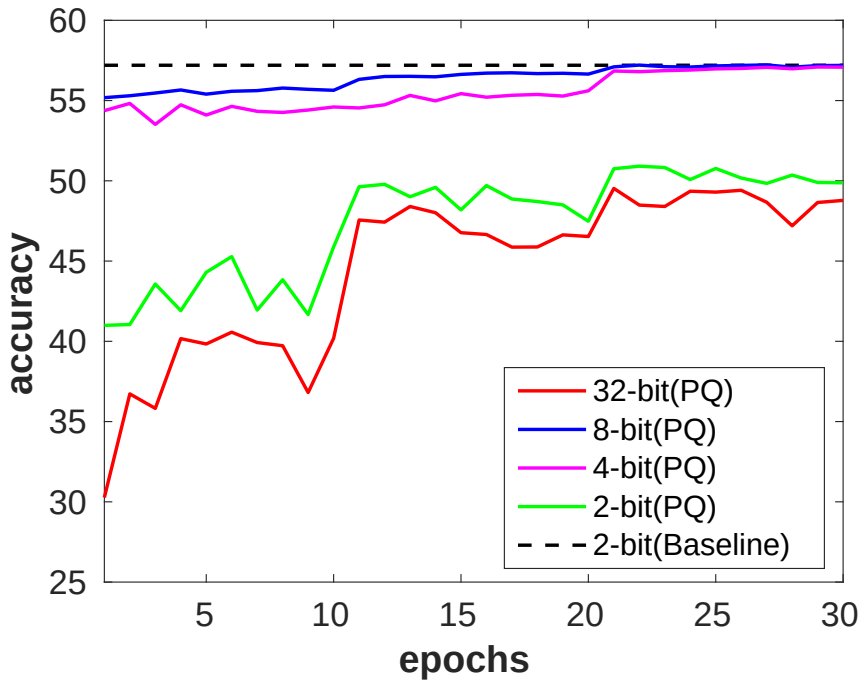


Figure 3.4: Validation accuracy of the progressive quantization approach using AlexNet on ImageNet.

The effect of the jointly guided training: We also investigate the effect of the guided joint training approach explained in Sec. 3.4.4. By comparing the results in Table 3.4 and Table 3.5, we can find that *Guided* method steadily improves the *baseline* method by a promising margin. This justifies the low-precision model can always benefit by learning from the full-precision model. What's more, we can find *PQ+TS+Guided* outperforms *PQ+TS* in all settings. This shows that the guided training strategy and the progressive learning mechanism can benefit from each other for further improvement.

Joint vs. without joint: We further illustrate the joint optimization effect on guided training in Figure 3.5. For explaning convenience, we implement it based on the method *Stage2+Guided* and report the 2-bit AlexNet top-1 validation accuracy on ImageNet. From the figure, we can observe that both the full-precision model and its low-precision counterpart can benefit from learning from each other. In contrast, if we keep the full-precision model unchanged, apparent performance drop is ob-

served. This result strongly supports our assumption that the high-precision and the low-precision models should be jointly optimized in order to obtain the optimal gradient during training. The improvement on the full-precision model may due to the ensemble learning with the low-precision model and similar observation is found in [Zhang et al., 2017d] but with different task.

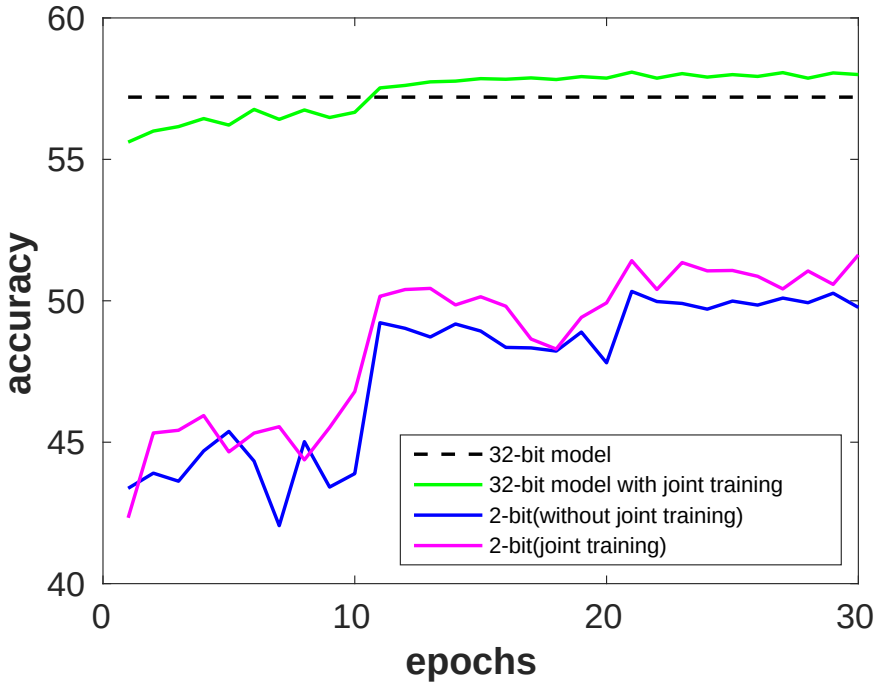


Figure 3.5: The effect of the joint training strategy using AlexNet on ImageNet.

3.6 Summary

In this chapter, we have proposed three novel approaches to solve the optimization problem for quantizing the network with both low-precision weights and activations. We first propose a two-stage approach to quantize the weights and activations in a two-step manner. We also observe that continuously quantizing from high-precision to low-precision is also beneficial to the final performance. We have shown that these two heuristics lead to better performance of low-precision networks. Furthermore, to better utilize the knowledge from the full-precision model, we have also proposed

joint learning of the low-precision model and its full-precision counterpart – this approach ensures that the full-precision model remains close to the low-precision approximation and regularizes the training optimization more effectively. We show that even using only 4-bit weights and activations for all layers, we can outperform the 32-bit model on ImageNet and Cifar100 with either AlexNet or ResNet-50.

Chapter 4

Fast Training of Triplet-based Deep Binary Embedding Networks

Statement of Authorship

Title of Paper	Fast training of Triplet-based deep binary embedding networks
Publication Status	[X] Published [ ] Accepted for Publication [ ] Submitted for Publication [ ] Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	Published at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

Principal Author

Name of Principal Author (Candidate)	Bohan Zhuang
Contribution to the Paper	Wrote the paper and completed the experiments.
Overall percentage (%)	90%
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.
Signature	Date 7/12/17

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Guosheng Lin
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/17

Name of Co-Author	Chunhua Shen
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/17

Please cut and paste additional co-author panels here as required.

Name of Co-Author	Ian Reid		
Contribution to the Paper	Helped design the method and modified the paper.		
Signature		Date	7/12/17

4.1 Overview

In this chapter, we aim to learn a mapping (or embedding) from images to a compact binary space in which Hamming distances correspond to a ranking measure for the image retrieval task.

We make use of a triplet loss because this has been shown to be most effective for ranking problems. However, training in previous works can be prohibitively expensive due to the fact that optimization is directly performed on the triplet space, where the number of possible triplets for training is cubic in the number of training examples. To address this issue, we propose to formulate high-order binary codes learning as a multi-label classification problem by explicitly separating learning into two interleaved stages. To solve the first stage, we design a large-scale high-order binary codes inference algorithm to reduce the high-order objective to a standard binary quadratic problem such that graph cuts can be used to efficiently infer the binary codes which serve as the labels of each training datum. In the second stage we propose to map the original image to compact binary codes via carefully designed deep convolutional neural networks (CNNs) and the hashing function fitting can be solved by training binary CNN classifiers. An incremental/interleaved optimization strategy is proffered to ensure that these two steps are interactive with each other during training for better accuracy. We conduct experiments on several benchmark datasets, which demonstrate both improved training time (by as much as two orders of magnitude) as well as producing state-of-the-art hashing for various retrieval tasks.

4.2 Introduction

With the rapid development of big data, large-scale nearest neighbor search with binary hash codes has attracted much more attention. Hashing methods aim to map the original features to compact binary codes that are able to preserve the *semantic* structure of the original features in the Hamming space. Compact binary codes are

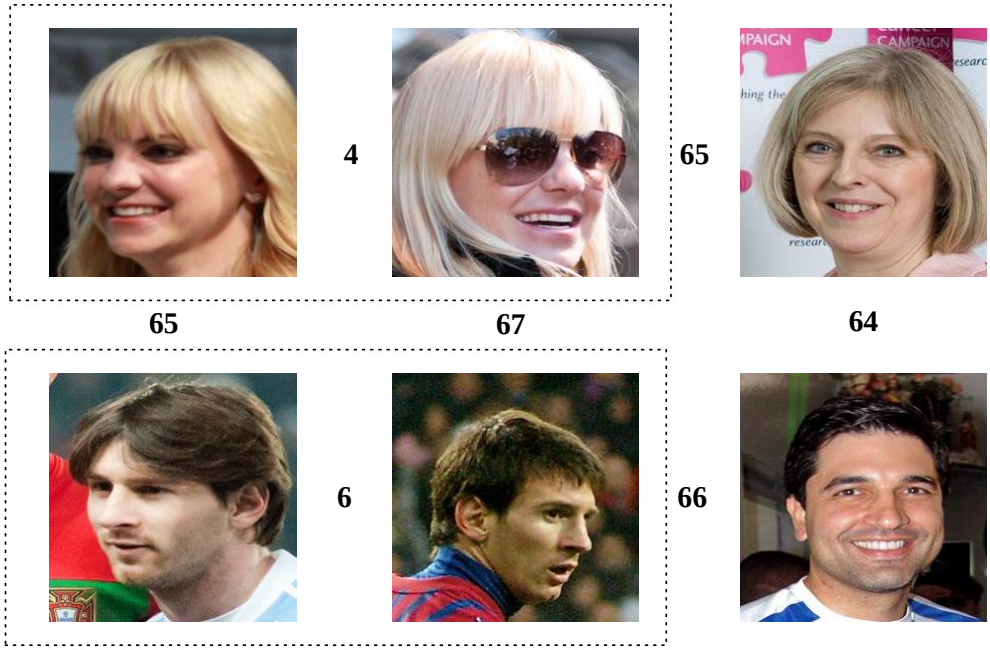


Figure 4.1: The Hamming distances calculated using the proposed hashing framework between pairs of faces. Each row represents a triplet of samples and the face pairs enclosed by a rectangle are from the same identity. Here each face image is represented by a 128-dimensional binary codes vector. We can see that a threshold of about 63 can correctly classify same-identity and different-identity pairs of faces.

extremely suitable for efficient data storage and fast search. A few hashing methods in the literature incorporate the triplet ranking loss to learn codes that preserve relative similarity relations [Norouzi et al., 2012; Lai et al., 2015; Zhao et al., 2015; Zhang et al., 2015; Li et al., 2013]. In these works usually a triplet ranking loss is defined, followed by solving an expensive optimization problem. For instance, Lai *et al.* [Lai et al., 2015] and Zhao *et al.* [Zhao et al., 2015] map original features into binary codes via deep convolutional neural networks (CNNs). Both use a triplet ranking loss designed to preserve relative similarities, with the key difference being in the exact form of the loss function used. Similarly, FaceNet [Schroff et al., 2015] uses the triplet loss to learn a real-valued compact embedding of faces. All these methods suffer from huge training complexity, because they directly train the CNNs using the triplets, the number of which scales cubically with the number of images in the training set. For example, the training of FaceNet [Schroff et al., 2015] took a few months on Google's computer clusters. Other work like [Wang et al., 2014] simply subsamples a small subset to reduce the computation complexity.

To address this issue, we employ a collaborative two-step approach, originally proposed in [Lin et al., 2013], to avoid directly learning hash functions based on the triplet ranking loss. This two-step approach enables us to convert triplet-based hashing into an efficient combination of solving binary quadratic programs and learning conventional CNN classifiers. Hence, we don't need to directly optimize the loss function with huge number of triplets to learn deep hash functions. The result is an algorithm with computational complexity that is orders of magnitude lower than existing work such as [Zhao et al., 2015; Schroff et al., 2015], but without sacrificing accuracy.

The two-step approach to hashing advocated by [Lin et al., 2014a, 2013] uses decision trees as hash functions in combination with the design of efficient binary code inference methods. The main difference of our work is as follows. The work in [Lin et al., 2014a, 2013] only preserves the *pairwise* similarity relations which do

not directly encode relative semantic similarity relationships that are important for ranking-based tasks. In contrast, we use a triplet-based ranking loss to preserve relative semantic relationships. However it is not trivial to extend the first step (binary code inference) in [Lin et al., 2014a] to triplet-based loss functions. The formulated binary quadratic problem (BQP) in [Lin et al., 2014a] can be viewed as a pairwise Markov random field (MRF) inference problem, while in our case we need to solve large-scale *high-order* MRF inference. We here propose an efficient high-order binary code inference algorithm, in which we equivalently convert the binary high-order inference into the second-order binary quadratic problem, and graph cuts based block search method can be applied. In the second step of hash function learning, the work of [Lin et al., 2014a, 2013] relies on training classifiers such as linear SVM or decision trees on handcrafted features. We instead fit deep CNNs with incremental optimization to simultaneously learn feature representations and hash codes.

Our contributions are summarized as follows.

- To address the issue of prohibitively high computational complexity in triplet-based binary code learning, we propose a new efficient and flexible framework for interactively inferring binary codes and learning the deep hash functions, using a triplet-based loss function. We show how to convert the high-order loss introduced by the triplets into a binary quadratic problem that can be optimized efficiently in the manner of [Lin et al., 2014a], using block-coordinate descent with graph-cuts. To learn the mapping from images to hash codes, we design deep CNNs capable of preserving their semantic ranking information of the data.
- We propose a novel incremental group-wise training approach, that interleaves finding groups of bits of the hash codes, with learning the hash functions. We show experimentally that this approach improves the quality of hash functions while retaining the advantage of efficient training.
- We demonstrate that our method outperforms many existing state-of-the-art

hashing methods on several benchmark datasets by a large margin. We also demonstrate our hashing method in the context of a face search/retrieval system. We achieve the best reported results on face search under the IJB-A protocol.

4.3 The proposed approach

Our general problem formulation is as follows. Let $\mathcal{D} = \{(i, j, k) \,|\, s(\mathbf{x}_i, \mathbf{x}_j) > s(\mathbf{x}_i, \mathbf{x}_k)\}$ be a set of training triplet samples, in which $s(\cdot, \cdot)$ is some semantic similarity measures, $\mathbf{x}_i$ is the i-th training sample and $\mathbf{x}_i$ is semantically more similar to $\mathbf{x}_j$ than to $\mathbf{x}_k$. Let $h(\mathbf{x}) \in \{-1, 1\}^q$ be the q-bit hash codes of image $\mathbf{x}$. We simplify the notation by rewriting $h(\mathbf{x}_i)$, $h(\mathbf{x}_j)$ and $h(\mathbf{x}_k)$ using z_i, z_j and z_k, respectively. Our goal is to learn embedding hash functions $h(\cdot)$ to preserve the relative similarity ranking order for the images after being mapped into the binary Hamming space. For that purpose, we define a general form of loss functions:

$$\min_{\mathbf{Z}} \sum_{(i,j,k)\in\mathcal{D}} \mathcal{L}(z_i, z_j, z_k), \text{s.t. } \mathbf{Z} \in \{-1, 1\}^{q\times n}. \tag{4.1}$$

Here $\mathbf{Z}$ is the matrix that collects binary codes for all the n data points and q is the bit length. $\mathcal{L}$ is a triplet loss function.

Unlike approaches such as [Zhao et al., 2015], our method shares the advantage of [Lin et al., 2013] that we are not tied to a specific form of the loss. One typical example of losses that could be used include the *Hinge ranking loss*:

$$\mathcal{L}(z_i, z_j, z_k) = \max(0,\, q/2 - (d_H(z_i, z_j) - d_H(z_i, z_k))). \tag{4.2}$$

Here $d_H(\cdot, \cdot)$ is the Hamming distance.

We propose an approach to learning binary hash codes that proceeds in two stages. The first stage uses the labelled training data to infer a set of binary codes in

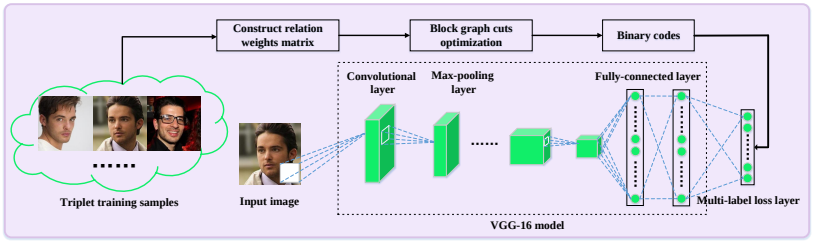


Figure 4.2: Overview of the proposed hashing framework for training one group of binary codes. The framework includes two steps: binary code inference and hash function learning with multi-label CNNs. The inferred binary codes are needed by the multi-label layer of the deep hash functions. The CNN structure of the first a few layers is same as the VGG-16 network.

which the hamming distance between codes preserves the semantic ranking between triplets of data. The second stage uses deep CNNs to learn the mapping from images to the binary code space (i.e. to learn the hash functions). A similar two-stage approach was advocated in [Lin et al., 2014a], but that work used only pairwise data, and used boosted decision trees rather than deep CNNs to learn the hash functions.

There are various difficulties associated with direct application of triplet losses, and of CNNs to the problem. First, the binary code learning stage requires optimization of Eq. (4.1) which is in general NP-hard. In Sec. 4.4, we describe how to infer binary codes with triplet ranking loss by reducing the problem to a binary quadratic program. The use of triplets considerably complicates this process and so this is one of our significant contributions in this chapter. Second, while the two-stage approach gains significantly in training time, it has the disadvantage that the learning of the codes and the hash functions do not interact and therefore cannot be mutually beneficial. We propose a method to interleave the code and hash function learning into groups of bits, a process that retains much of the training efficiency, but improves the quality of the codes and hash functions considerably. We explain our use of CNNs and this interleaved and incremental learning in Sec. 4.5 below.

4.4 Inference for binary codes with triplet ranking loss

Since simultaneously infer multiple bits are intractable in inference task, inspired by the work of [Lin et al., 2014a], we sequentially solve for one bit at a time conditioning on previous bits. When solving for the r-th bit, the previous $r-1$ bits are fixed. The binary inference problem becomes minimization of the following objective:

$$\begin{aligned}\sum_{(i,j,k)\in\mathcal{D}} \mathcal{L}(z_{r,i}, z_{r,j}, z_{r,k}; \boldsymbol{z}_i^{(r-1)}, \boldsymbol{z}_j^{(r-1)}, \boldsymbol{z}_k^{(r-1)}),\\ = \sum_{(i,j,k)\in\mathcal{D}} \ell_r(z_{r,i}, z_{r,j}, z_{r,k}),\end{aligned} \tag{4.3}$$

where ℓ_r is the loss function output of the r-th bit conditioned on the previous bits. $z_{r,i}$ is the binary code of the i-th data point and the r-th bit, $\boldsymbol{z}_i^{(r-1)}$ is the binary code vector of the previous $r-1$ bits for the i-th data point.

4.4.1 Solving high-order binary inference problem

Directly optimizing the loss function which involves high-order relations (more than pairwise relations) in Eq. (4.3) is difficult since the optimization involves an extremely large number of triplets, and so can be computationally intractable. To address this problem, we show here how to convert the high-order inference task to a second-order problem which is much more feasible to be optimized. The key "special properties" of the binary space that we rely on are: (i) the possibility of enumerating all possible inputs (there are $2^3 = 8$); (ii) the symmetry of the hamming distance $d(.,.)$. Based on this, the triplet loss can be decomposed into a set of second-order combinations as:

$$\begin{aligned}\ell_r(z_{r,i}, z_{r,j}, z_{r,k}) = \alpha_{ii} z_{r,i} z_{r,i} + \alpha_{ij} z_{r,i} z_{r,j} + \alpha_{ik} z_{r,i} z_{r,k}\\ + \alpha_{ji} z_{r,j} z_{r,i} + \alpha_{jj} z_{r,j} z_{r,j} + \alpha_{jk} z_{r,j} z_{r,k} + \alpha_{ki} z_{r,k} z_{r,i}\\ + \alpha_{kj} z_{r,k} z_{r,j} + \alpha_{kk} z_{r,k} z_{r,k},\end{aligned} \tag{4.4}$$

where $\alpha_{..}$ are the coefficients of the corresponding second-order combinations. Then we will show that there exists a solution for $\boldsymbol{\alpha}$ to make it a valid decomposition. Here we ignore the redundant terms in Eq. (4.4), hence it can be rewritten as

$$\begin{aligned}\ell_r(z_{r,i}, z_{r,j}, z_{r,k}) &= \alpha_{ii} z_{r,i} z_{r,i} + \alpha_{ij} z_{r,i} z_{r,j} \\ &\quad + \alpha_{ik} z_{r,i} z_{r,k} + \alpha_{jk} z_{r,j} z_{r,k} = \boldsymbol{\alpha}^{\mathrm{T}} \mathbf{v}, \end{aligned} \tag{4.5}$$

$$\begin{aligned}\text{where,} \quad \boldsymbol{\alpha} &= [\alpha_{ii}, \alpha_{ij}, \alpha_{ik}, \alpha_{jk}], \\ \mathbf{v} &= [z_{r,i} z_{r,i}, z_{r,i} z_{r,j}, z_{r,i} z_{r,k}, z_{r,j} z_{r,k}].\end{aligned}$$

ℓ_r has 8 possible input combinations for $(z_{r,i}, z_{r,j}, z_{r,k})$ (or equivalently $\mathbf{v}$ has 8 possible value combinations), leading to 8 constraints of the form of (4.5). Because the loss is defined on Hamming distance/affinity, changing the sign of every input leads to identical value of the loss, thus some of these combinations lead to redundant constraints. Eliminating all these redundant combinations leaves only four independent equations (4.5). Stacking these so that each $\mathbf{v}$ forms a row of a matrix yields the follow set of equations:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \boldsymbol{\alpha} = \begin{bmatrix} \ell_r(1,1,1) \\ \ell_r(1,1,-1) \\ \ell_r(1,-1,1) \\ \ell_r(1,-1,-1) \end{bmatrix}. \tag{4.6}$$

which can be easily inverted to yield the unique solution of $\boldsymbol{\alpha}$. This shows that for a given triplet loss function, we can decompose it into a set of pairwise terms for each triplet.

We now seek a solution for $\boldsymbol{z}_{(r)}$ – the r^{th} bit of the code for every data point – that optimizes the triplet relations. Because the triplet relations are now encoded as

Algorithm 4: Greedy method for constructing blocks

Input: Training images: $\{\mathbf{x}_1, ...\mathbf{x}_n\}$; Relation weights matrix: $\mathbf{W}$.
Output: Sub-modular blocks: $\{\mathcal{S}_1, \mathcal{S}_2, ...\}$.

```
U ← {x_1, ..., x_n}; t = 0;
while U ≠ ∅ do
    t = t + 1; S_t ← ∅; choose an arbitrary x_i from U;
    Let H be U ∪ {x_j | w_ij < 0}
    for each x_j in H do
        if w_jk ≤ 0 for k = 1, 2, ..., |S_t| then
            Add x_j to S_t; If x_j ∈ U, remove it;
```

pairwise relations, we can solve for $\boldsymbol{z}_{(r)}$ as follows. We define $\mathbf{W} \in R^{n \times n}$ as a weight matrix in which (i, j)-th element of $\mathbf{W}$, w_{ij}, represents a relation weight between the i-th and j-th training points. Specifically, each element of $\mathbf{W}$ is computed as

$$w_{ij} = \sum_{\forall(i,j)} \alpha_{ij}, \tag{4.7}$$

where α_{ij} are the coefficients corresponding to the pair (i, j). There will be one such α_{ij} for every triplet in which data points $\mathbf{x}_i$ and $\mathbf{x}_j$ appear.

The triplet optimization problem in Eq. (4.3) can now be equivalently formulated as

$$\min_{\boldsymbol{z}_{(r)} \in \{-1,1\}^n} \boldsymbol{z}_{(r)}^{\mathrm{T}} \mathbf{W} \boldsymbol{z}_{(r)}. \tag{4.8}$$

Note that the coefficients matrix $\mathbf{W}$ is sparse and symmetric, therefore Eq. (4.8) is a standard binary quadratic problem. Although we have now shown how to convert the third-order objective in Eq. (4.3) into a second-order formulation amenable to BQP, a further issue remains: the quadratic objective above contains non-submodular terms, and is therefore difficult to optimize.

To address this, we follow the proposal in [Lin et al., 2014a]. This proceeds by creating a set of sub-problems (or "blocks") each involving a subset of the variables $\boldsymbol{z}_{(r)}$ in which the pairwise relations are all sub-modular. The sub-problems are then solved in turn, treating the variables that are not involved in the current block as

Algorithm 5: Two-step approach for learning deep binary embedding networks

Input: Training images: $\{\mathbf{x}_1, ...\mathbf{x}_n\}$; Relation map: $\mathbf{M}$; group length: a; number of groups: b.
Output: The deep hash functions: $h(\cdot)$.

```
for i = 1, ...b do
    for j = 1, ...a do
        Solve linear equations to construct the relation weight matrix W;
        Apply Block Graph-Cut algorithm [Lin et al., 2014a] to solve
        ((i − 1) × a + j)-th bit hash codes;
    Learn the deep hash functions h(·) based on i × a bits hash codes;
    Simultaneously update i × a bits hash codes by the output of h(·).
```

constants. The inference problem for one block is written as

$$\min_{z_r \in \{-1,1\}^n} \sum_{i \in \mathcal{S}} u_i z_{r,i} + \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} v_{ij} z_{r,i} z_{r,j}, \tag{4.9}$$

$$\text{where, } u_i = 2 \sum_{j \notin \mathcal{S}} w_{ij} z_{r,j}, \; v_{ij} = w_{ij},$$

and $\mathcal{S}$ is the block to be optimized. Since the above inference problem for one block is sub-modular, we can solve it efficiently using graph cuts.

Algorithm (4) details how the blocks are defined. It is subtly different from [Lin et al., 2014a]; because we are using a triplet loss, the criterion for inclusion in a block is to ensure $w_{ij} < 0$ for each pair $\mathbf{x}_i, \mathbf{x}_j$ in the block, which guarantees sub-modularity for all pairs.

4.4.2 Loss function

The discussion above provides a general framework for learning the binary codes using a triplet loss, but is agnostic to the exact form of the loss. In the experiments reported in this chapter, we use ℓ_r as the triplet-based hinge loss function defined in Eq. (4.2):

$$\ell_r(...) = \max(0, \; r/2 - \Delta d_H^{(r-1)} - \Delta d_H^r), \tag{4.10}$$

where,

$$\Delta d_H^{(r-1)} = d_H(\boldsymbol{z}_i^{(r-1)}, \boldsymbol{z}_j^{(r-1)}) - d_H(\boldsymbol{z}_i^{(r-1)}, \boldsymbol{z}_k^{(r-1)}),$$
$$\Delta d_H^{r} = d_H(z_{r,i}, z_{r,j}) - d_H(z_{r,i}, z_{r,k}).$$

4.5 Deep hash functions learning

Our general scheme now requires that we learn hash functions $h(.)$ that map from data points $\mathbf{x}_i$ to binary codes. We propose to do this using deep CNNs because they have repeatedly been shown to be very effective for similar tasks. The straightforward approach is then to use the training samples, and their known codes as the labelled training set for a standard CNN. As we have noted this two-stage approach yields significant training time gains.

However a major disadvantage is that because the binary codes are determined independently of the hash functions, and the hash functions have no possibility to influence the choice of binary codes. Ideally these stages would interact so that the choice of binary hash codes is influenced not only by the ground-truth relative similarity relations but also by how hard the training points are.

To address this, we propose an interleaved process where we infer a group of bits within a code, followed by learning suitable hash functions for that set of bits and its predecessors, followed in turn by inference of the next group of bits, and so on. This provides a compromise between independently learning the codes and hash functions, and a more end-to-end – but very expensive – approach such as [Lai et al., 2015].

4.5.1 Incremental optimization

Our key idea here is to optimize the hashing framework in an incremental group-wise manner. More specifically, we assume there are b groups of bits and each group has a bits (e.g., for 64-bit codes we may break this into 8 groups of 8 bits each). For

convenience, we shall refer to inference of the p-th group binary codes followed by learning the deep hash functions, as the "p-th training stage". In the p-th training stage, we first infer the a bits of the p-th group one bit at a time (as described in Sec. 4.4) and then train the network parameters θ so that it minimizes the cross-entropy loss:

$$-\sum_{\rho=1}^{r}\sum_{i=1}^{n}[\delta(z_{\rho,i}=1)\log z'_{\rho,i}+\delta(z_{\rho,i}=-1)\log(1-z'_{\rho,i})], \tag{4.11}$$

where $\delta(\cdot)$ is the indication function. Here at the p-th stage we are targetting the first $r = pa$ bits of the code; $z'_{\rho,i}$ is the ρ-th output of the last sigmoid layer for the i-th training sample; $z_{\rho,i}$ is the corresponding bit of the binary code obtained from the inference step which serves as the target label of the multi-label classification problem above. Note that in the p-th training stage, the bits from all p groups are used to guide the learning of the deep hash functions.

Having completed training the hash functions, we then update the binary codes for all p groups by the output of the learned hash functions. The effect of this is to ensure that the error in the learned hash functions will influence the inference of the next group of hash bits.

This incremental training approach adaptively regulates the binary codes according to both the fitting capability of the deep hash functions and the properties of the training data, steadily improving the quality of hash codes and the final performance. Finally, we summarize our hashing framework in Algorithm 5.

4.5.2 Network architecture

The network of learning deep hash functions consists of multiple convolutional, pooling, and fully connected layers (we follow the VGG-16 model), and a multi-label loss layer for multi-label classification.

We use the pre-trained VGG-16 [Simonyan and Zisserman, 2015] model for initialization, which is trained on the large-scale ImageNet dataset. The multiple convolution-pooling and fully connected layers are used to capture mid-level image representa-

tions. The intermediate output of the last fully connected layer are mapped to a multi-label layer as the feature representation. Then neurons in the multi-label layer are activated by a sigmoid function so that the activations are approximated to $[0, 1]$, followed by the cross-entropy loss of Eq. (4.11) for multi-label classification.

4.6 Experiments

Experimental settings We test the proposed hashing method on two multi-class datasets, one multi-label dataset and one face retrieval dataset. For multi-class datasets, we use the MIT Indoor dataset [Quattoni and Torralba, 2009] and CIFAR-10 dataset [Krizhevsky, 2009]. The MIT Indoor dataset contains 67 indoor scene categories, and 6,700 images for evaluation. CIFAR-10 contains 60,000 small images in 10 classes. For multilevel similarity measurement, we test our method on the multi-label dataset NUS-WIDE [Chua et al., 2009]. The NUS-WIDE dataset is a large database containing 269,648 images annotated with 81 concepts. We compare the search accuracies with four recent state-of-the-art state-of-the-art hashing methods, including SFHC [Lai et al., 2015] (the recent deep CNNs method), FSH [Lin et al., 2014a] (two-step hashing approach using decision trees), KSH [Liu et al., 2012] and ITQ [Gong et al., 2013].

For fair comparison, we evaluate the compared hashing methods FSH, KSH and ITQ on the features obtained from the activations of the last hidden layer of the VGG-16 model pre-trained on the ImageNet ILSVRC-2012 dataset [Russakovsky et al., 2015a]. We find that using deep CNN features in general improve the performance for these three hashing methods, compared with what was originally proposed. We initialize our CNN using the pre-trained model and fine-tune the network on the corresponding training set.

Again for fair comparison, for the deep CNN approach SFHC, we replace its network structure (convolution-pooling, fully-connected layers) with the VGG-16 model and end-to-end train the network based on the triplet hinge loss used in the original

paper. We implement SFHC using *Theano* [Bastien et al., 2012] and train the model using two GeForce GTX Titan X. The triplet samples are randomly generated in the course of training, following [Lai et al., 2015].

For the NUS-WIDE dataset, we construct two comparison settings, setting-1 and setting-2. For setting-1, following the previous work [Lai et al., 2015; Liu et al., 2011], we consider the 21 most frequent tags and the similarity is defined based on whether two images share at least one common tag. For setting-2, we use the similarity precision evaluation metric to evaluate pairwise and triplet performance. As in [Wang et al., 2014], similarity precision is defined as the % of triplets being correctly ranked.

Given a triplet image set $(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$, where $s(\mathbf{x}_i, \mathbf{x}_j) > s(\mathbf{x}_i, \mathbf{x}_k)$. We assume $\mathbf{x}_i$ as the query, if the rank of $\mathbf{x}_j$ is higher than $\mathbf{x}_k$, then we say triplet is correctly ranked. We first randomly sample 1000 probe images from all the data sharing the selected 21 attributes in setting-1. Then we obtain a ranking list for each probe image according to how many attributes it shares with the data and randomly generate 50 triplets per probe image according to the ranking list to form the test set. For the triplet-based methods, the sampled training data is the same as in setting-1. For the compared pairwise-based methods, we directly use the hash functions learned in setting-1 since semantic ranking information cannot be incorporated into the pairwise-based inference pipeline. For CIFAR-10 and NUS-WIDE setting-1, we use the same experimental setting as described in [Lai et al., 2015].

We use two evaluation metrics: Mean Average Precision (MAP) and the precision of the top-K retrieved examples (Precision), where K is set to 100 in CIFAR-10 and NUS-WIDE setting-1 and set to 80 in MIT Indoor dataset. For NUS-WIDE setting-1, we calculate the MAP values within the top 5000 returned neighbors. The results are represented in Figure 4.3 and Figure 4.4.

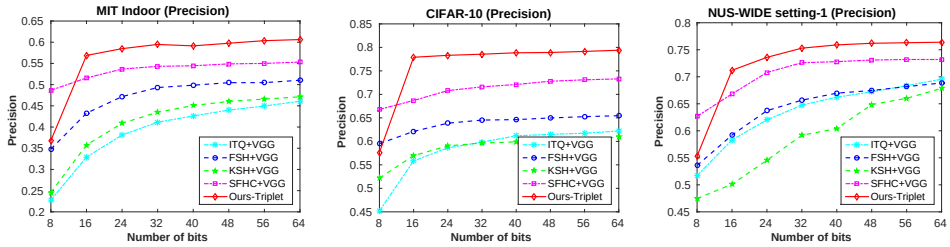


Figure 4.3: The precision curves on three datasets. We compare several state-of-the-art algorithms including ITQ [Gong et al., 2013], KSH [Liu et al., 2012], FSH [Lin et al., 2014a] with features extracted from VGG-16 model which is fine-tuned on the corresponding training set and SHFC [Lai et al., 2015] which is implemented using the VGG-16 network structure.

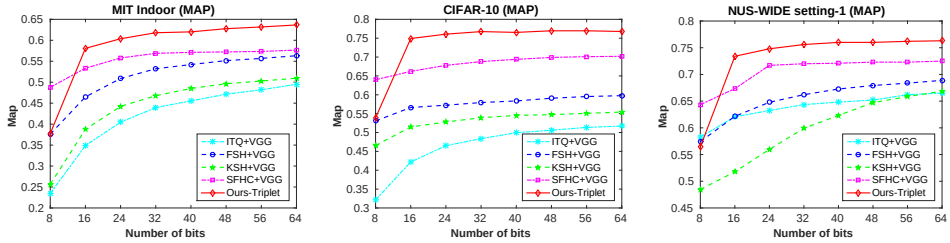


Figure 4.4: The mean average precision curves on three datasets. Settings are the same as in Figure 4.3.

4.6.1 Implementation details

We implement the network training based on the CNN toolbox *Theano*. Training is done on a standard desktop with a GeForce GTX TITAN X with 12GB memory. In all experiments, we set the mini-batch size for gradient descent to 50, momentum 0.9, weight decay 0.0005 and dropout rate 0.5 on the fully connected layer to avoid over-fitting. The number of binary codes per group is set to 8.

4.6.2 Analysis of retrieval results

On all the three datasets, our proposed method shows superior performance in terms of MAP and precision evaluation metrics against the most related work SFHC (deep CNN) and FSH (two-step hashing with boosted trees). As expected, the training speed of our method is much faster than SFHC, and the result is summarized in Table 4.1. Rather than simply end-to-end learn the hash functions, our method incor-

Figure 4.5: The similarity precision curves on NUS-WIDE setting-2.

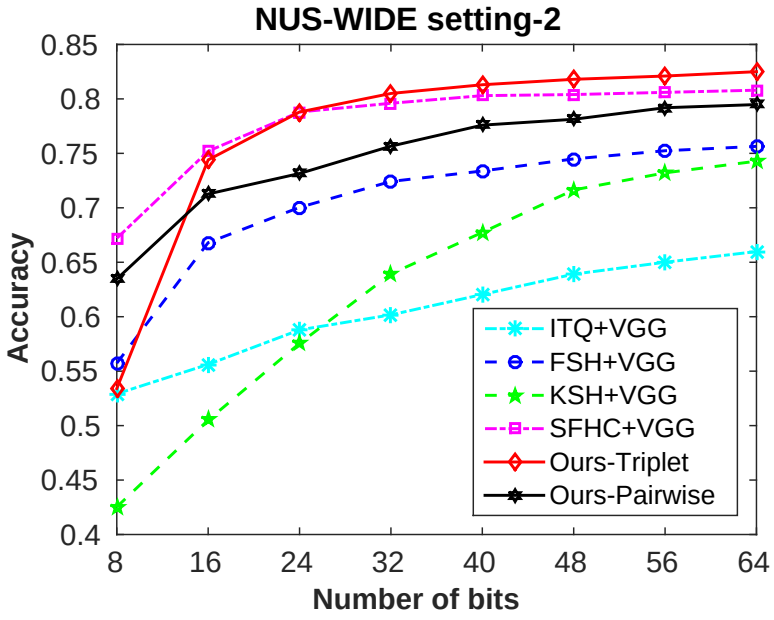
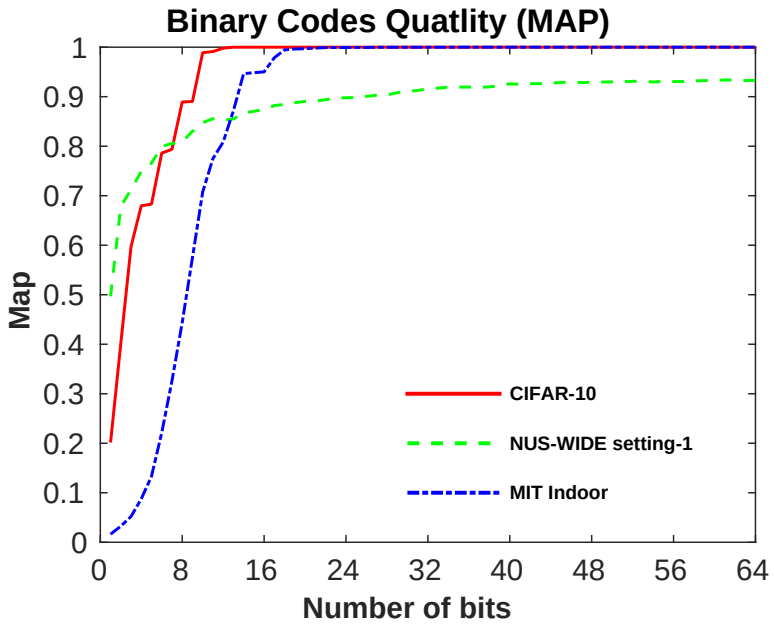


Figure 4.6: Evaluation of the inference performance on three datasets.



porates hash functions learning with a collaborative inference step, where the image representation learning and hash coding can benefit each other through this feedback scheme.

Compared to FSH, the results demonstrate the effectiveness of incorporating relative similarity information as supervision. Note that FSH is based on pairwise information while ours uses triplet based ranking information to learn hash codes. The triplet loss may be better for retrieval tasks because it is directly linked to retrieval measure such as the AUC score. The pairwise loss used by FSH encourages all images in one category to be projected onto a single point in the Hamming space. The triplet loss maximizes a margin between each pair of same-category images and images from different categories. As argued in [Schroff et al., 2015; Weinberger and Saul, 2009], this may enable images belonging to the same category to reside on a manifold; and at the same time to maintain a distance from other categories.

Table 4.1: Training time of the proposed method and the method SFHC [Lai et al., 2015] on three datasets. In terms of training time, our method is significantly faster than SFHC.

Method	Training Time (hours)			Number of GPUs
	MIT Indoor	CIFAR-10	NUS-WIDE setting-1	
Ours-Triplet	18	15	32	1
SFHC	186	174	365	2

4.6.3 Triplet vs. pairwise

From the results shown in Figure 4.5, we can clearly observe the superiority of triplet-based methods on the ranking based evaluation metric. Thanks to the high quality binary codes and the strong fitting capability of our deep model, our proposed method provides much better performance than pairwise methods by a large margin.

Since the two triplet-based methods (Ours-Triplet and SFHC) simultaneously learn feature representations and hash codes while considering the semantic ranking information, they are more likely to learn hash functions that are tailored for the ranking-based retrieval metric than the pairwise-based methods (Ours-pairwise and

FSH).

4.6.4 Evaluation of binary codes quality

Table 4.2: Face search accuracies under the IJB-A protocol. Results for GOTS and OpenBR are quoted from [Klare et al., 2015]. Results are reported as the average $\pm$ standard deviation over the 10-fold cross validation sets specified in the IJB-A protocol.

Algorithm	CMC (closed-set search)		FNIR @ FPIR (open-set search)	
	Rank-1	Rank-5	0.1	0.01
GORS	0.443 ± 0.021	0.595 ± 0.020	0.765 ± 0.033	0.953 ± 0.024
OpenBR	0.246 ± 0.011	0.375 ± 0.008	0.851 ± 0.028	0.934 ± 0.017
Deep Face Search[Wang et al., 2015a]	0.820 ± 0.024	0.929 ± 0.013	0.387 ± 0.032	0.617 ± 0.063
Proposed Method	$\mathbf{0.831 \pm 0.020}$	$\mathbf{0.937 \pm 0.015}$	$\mathbf{0.369 \pm 0.028}$	$\mathbf{0.598 \pm 0.048}$

We evaluate the binary codes quality on CIFAR-10, MIT Indoor and NUS-WIDE setting-1 datasets (see Figure 4.6). To evaluate the effectiveness of the binary codes inference pipeline, we infer 64 binary bits without learning the deep hash functions. Then the training database is used as both the probe set and the gallery set for evaluating the inference performance. For the three datasets, we calculate the MAP values within the returned neighbors. We can observe that for CIFAR-10, the binary codes converge very fast at around 10-th bits. MIT Indoor dataset converges slightly slower due to the fact that it has more classes. The binary codes can still perfectly separate all the training samples from different classes. This is because the relations between training points are very simple due to the multi-class similarity relationships. In contrast, due to the complicated relationships between the multi-label training samples, the accuracy of NUS-WIDE setting-1 keeps improving up to 64 bits and is lower than those multi-class datasets. We can see that the code quality is directly proportional to the final retrieval performance. This makes sense since the deep hash functions are learned to fit the binary codes, so the performance of the inference pipeline has a direct impact on the quality of the learned deep hash functions.

4.6.5 Face retrieval

We implement the face search application as follows. *Data preprocessing.* The preprocessing pipeline is: 1) detect the face region using the robust face detector [Mathias et al., 2014] and find 68 face landmarks using the (state-of-the-art) face alignment algorithm [Xiong and De la Torre, 2013]; 2) select the middle landmark between two eyes and the middle landmark of the mouth as alignment-anchor points, and align/scale the face image such that distance between the landmarks is 40 pixels; 3) finally we crop a 160×160 region around the mid-point of the two landmarks in (2).

Table 4.3: Face search accuracies of the proposed method under the IJB-A protocol using different bits per group.

Group length	CMC (closed-set search)		FNIR @ FPIR (open-set search)	
	Rank-1	Rank-5	0.1	0.01
8 bits	$\mathbf{0.831 \pm 0.020}$	$\mathbf{0.937 \pm 0.015}$	$\mathbf{0.369 \pm 0.028}$	$\mathbf{0.598 \pm 0.048}$
32 bits	0.818 ± 0.023	0.920 ± 0.016	0.385 ± 0.030	0.612 ± 0.052
64 bits	0.793 ± 0.024	0.908 ± 0.018	0.398 ± 0.036	0.627 ± 0.061
128 bits	0.778 ± 0.023	0.889 ± 0.020	0.415 ± 0.035	0.645 ± 0.058

Supervised pre-training. We pre-train the VGG-16 [Simonyan and Zisserman, 2015] network (using *Caffe* [Jia et al., 2014]) to classify all the 10575 subjects in the CASIA dataset [Yi et al., 2014]. This dataset has 494414 images of the 10575 subjects, and we double the number of training examples by horiozontal mirroring, making the feature representation more robust to pose variation.

We test the pre-trained model's discriminative power on the LFW verification data as follows. We use the last 4096-dimensional fully-connected layer as the feature representation and then use PCA to compress it into a 160-dimensional feature vector. Then CNN features are centered and normalized for evaluation. Under the standard LFW [Huang et al., 2007] face verification protocol, for a single network using only cosine similarity, we achieve an accuracy of $\mathbf{97.03}\% \pm \mathbf{0.98}\%$. Using the joint Bayesian method [Chen et al., 2012] for face verification, we achieve an accuracy of $\mathbf{98.18}\% \pm \mathbf{0.96}\%$.

Despite using only publicly available training data and one single network, the

performance of this model is competitive with state-of-the-art [Schroff et al., 2015; Taigman et al., 2014; Yi et al., 2014; Sun et al., 2015].

Face search. We then use the above pre-trained CNN model to initialize the deep CNN that models the hash functions of our proposed hashing method. We test the face search performance on the IARPA Janus Benchmark-A (IJB-A) dataset [Klare et al., 2015] which contains 500 subjects with a total of 25,813 face images. This dataset contains many challenging face images and defines both verification and search protocols. The search task (1:N search) is defined in terms of comparisons between templates consisting of several face images, rather than single face images. For the search protocol, which evaluates both closed-set and open-set search performance, 10-fold cross validation sets are defined based on both the probe and gallery sets consisting of templates. Given an image from the IJB-A dataset, we first detect and align the face following the data preprocessing pipeline. After processing, the final training set consists approximately 1 million faces and 1 billion randomly sampled triplets. Clearly, such a large-scale training dataset may render most existing triplet-based hashing methods computationally intractable. The deep hash functions are learned based on the proposed two-step hashing framework. After the deep hash functions are learned, we generate 128 bits hash codes for each input face image for fast face retrieval. The definitions of CMC, FNIR and FPIR are explained in [Wang et al., 2015a; Klare et al., 2015]. The results of the proposed method along with the compared algorithms are reported in Table 4.2. In [Wang et al., 2015a], a face is represented by the combined features extracted by 6 deep models. However, in our thesis, 128 bits binary codes are directed extracted by a single deep model for face representation which enjoys both faster searching speed and less storage space. Also, although using the same training database, the searching accuracy on two protocols both demonstrate the effectiveness of our hashing framework.

4.6.6 Evaluation of the incremental learning

We evaluate different group lengths used in the incremental learning to prove the effectiveness of such an optimization strategy. We implement the experiments on the face retrieval task as described above since there are sufficient training examples and faces are difficult for the deep architecture to fit because of the relatively weak discriminative information they share. The results are reported in Table 4.3. From the results, we clearly see that smaller group length corresponds to better search accuracies, demonstrating our assertion that incremental optimization helps in terms of code quality and the final performance.

4.7 Conclusion

In this chapter, we develop a general supervised hashing method with triplet ranking loss for large-scale image retrieval. Instead of directly training on the extremely large amount of triplet samples, we formulate learning of the deep hash functions as a multi-label classification problem, which allows us to learn deep hash functions orders of magnitude faster than the previous triplet based hashing methods in terms of training speed. The deep hash functions are learned in an incremental scheme, where the inferred binary codes are used to learn image representations and the learned hash functions can give feedback for boosting the quality of binary codes. Experiments demonstrate that the superiority of the proposed method over other state-of-the-art hashing methods.

Chapter 5

Attend in groups: a weakly-supervised deep learning framework for learning from web data

Statement of Authorship

Title of Paper	Attend in groups: a weakly-supervised deep learning framework for learning from web data
Publication Status	☒ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	Published at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

Principal Author

Name of Principal Author (Candidate)	Bohan Zhuang
Contribution to the Paper	Wrote the paper and completed the experiments.
Overall percentage (%)	90%
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.
Signature	Date: 7/12/2017

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Lingqiao Liu
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date: 7/12/2017

Name of Co-Author	Yao li
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date: 7/12/2017

Please cut and paste additional co-author panels here as required.

Name of Co-Author	Chunhua Shen
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/2017

Name of Co-Author	Ian Reid
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/17

5.1 Overview

Large-scale datasets have driven the rapid development of deep neural networks for visual recognition. However, annotating a massive dataset is expensive and time-consuming. Web images and their labels are, in comparison, much easier to obtain, but direct training on such automatically harvested images can lead to unsatisfactory performance, because the noisy labels of Web images adversely affect the learned recognition models. To address this drawback we propose an end-to-end weakly-supervised deep learning framework which is robust to the label noise in Web images. The proposed framework relies on two unified strategies – random grouping and attention – to effectively reduce the negative impact of noisy web image annotations. Specifically, random grouping stacks multiple images into a single training instance and thus increases the labeling accuracy at the instance level. Attention, on the other hand, suppresses the noisy signals from both incorrectly labeled images and less discriminative image regions. By conducting intensive experiments on two challenging datasets, including a newly collected fine-grained dataset with Web images of different car models, the superior performance of the proposed methods over competitive baselines is clearly demonstrated.

5.2 Introduction

Recent development of deep convolutional neural networks (CNNs) has led to great success in a variety of tasks including image classification [Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016a], object detection [Girshick et al., 2014; Ren et al., 2015; Liu et al., 2016], semantic segmentation [Long et al., 2015; Lin et al., 2016b] and others. This success is largely driven by the availability of large-scale well-annotated image datasets, e.g. ImageNet [Russakovsky et al., 2015a], MS COCO [Lin et al., 2014b] and PASCAL VOC [Everingham et al., 2010]. However, annotating a massive number of images is extremely labor-intensive and costly. To

reduce the annotating labor cost, an alternative approach is to obtain the image annotations directly from the image search engine from the Internet, e.g. Google image search or Bing images.

Web-scale image search engine mostly uses keywords as queries and the connection between keywords and images is established by the co-occurrence between the Web image and its surrounding text. Thus, the annotations of Web images returned by a search engine will be inevitably noisy since the query keywords may not be consistent with the visual content of target images. For example, using "black swan" as a query keyword, the retrieved images may contain "white swan," "swan painting" and some other different categories. These noisy labels can be misleading if we use them to train a classifier to learn the corresponding visual concept.

To overcome this drawback, we propose a deep learning framework designed to be more robust to the labeling noise and thus better able to leverage Web images for training. There are two key strategies in our framework: random grouping and attention. As will be shown later, these two strategies seamlessly work together to reduce the negative impact of label noise.

Specifically, the random grouping strategy randomly samples a few images and merges them into a single training instance. The idea is that although the probability of sampling an incorrectly labeled Web image is high, the probability of sampling an incorrectly labeled group is low because as long as one image in the group is correctly labeled, the label of the group is deemed correct (bag label as in multi-instance learning). In the proposed approach, each image is represented by the extracted contextual features depicting the visual patterns of local image regions. After the random grouping, a training instance is represented as the union of convolutional feature maps extracted from each image in the group. If there are any incorrectly labeled images in the group, the unified feature maps of an instance will contain a substantial amount of local features which are irrelevant to the group-level class annotation. To avoid the distraction of those local features, we apply the second

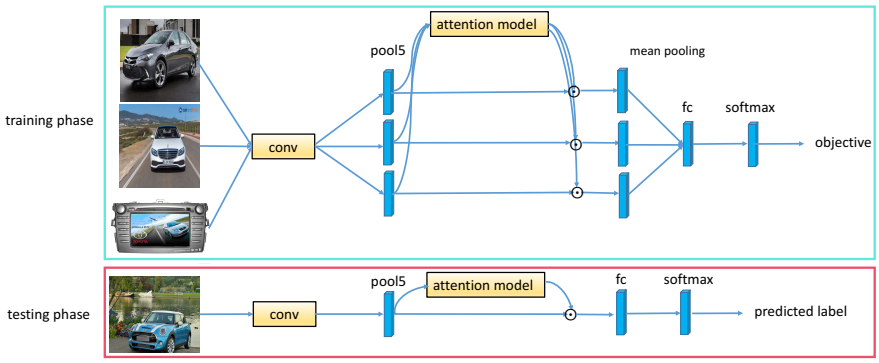


Figure 5.1: Overview of our "webly"-supervised learning pipeline. For the training phase, inputs are a group of images, including one correctly labeled image and two noise images from top to bottom. The convolutional layers are shared. The attention model is added on each training data and followed by a global average pooling layer to get the aggregated group-level representation, followed by a softmax layer for classification. For the testing phase, the input is a single image and output is the predicted class label.

strategy of our framework, the attention mechanism, to encourage the network not to focus on the irrelevant features.

To experimentally validate the robustness of the proposed method, we collect a large-scale car dataset using a Web image search engine. This dataset is particularly challenging due to its fine-grained nature. By conducting an experimental comparison on this dataset, we demonstrate that the proposed method achieves significantly better performance than competitive approaches.

5.3 Method

In our task, we intend to distill useful visual knowledge from the noisy Web data. It consists of correctly labeled samples and mislabeled samples on the Web. To make the classifier robust to noisy labels, we propose a deep learning framework by incorporating two strategies, random group training, and attention. The overview of our method is shown in Figure 5.1. At the training stage, we randomly group multiple training images into a single training instance as the input of our neural network. The proposed neural network architecture has two parts. The first part is similar to a

standard convolutional neural network which is comprised of multiple convolutional layers and pooling layers. The second part is an attentional pooling layer which selects parts of the neuron activations and pools the activations into the instance-level representation. Once the neural network is trained, we can drop off the random grouping module and takes a single image as input at the test stage.

In the following sections, we will elaborate the random grouping training and the attention module and discuss their benefits for reducing the impact of noisy labels.

5.3.1 Random grouping training

Random grouping training (RGT) aims at reducing the probability of sampling an incorrectly labeled instance and thus mitigate the risk confusing a neural work with wrong annotations. The idea of RGT is to stack multiple images of one class into a single grouped training instance of the same class. In practice, we implement this idea by stacking the last layer convolutional feature maps obtained from each image into a unified convolutional feature map and perform (attention based) pooling on this feature map to obtain the instance-level representation. In this sense, we can view the input of a grouped instance as a "merged image" and as long as one image is correctly labeled as containing the object-of-interest, the "merged image" indeed contains it. In other words, the grouped training instance is correctly labeled as long as one image within is correctly labeled.

Consequently, if the probability of sampling an incorrectly labeled image is ξ, then the probability of sampling a correctly labeled grouped instance will become

$$p = 1 - \xi^K \tag{5.1}$$

where K is the group size and when K becomes larger, the probability of sampling a correctly labeled instance will become very high. For example, if $\xi = 0.2$ and $K = 3$, p will be greater than 99%. However, when K becomes larger, the independence between multiple training instances will reduce and this tends to undermine the

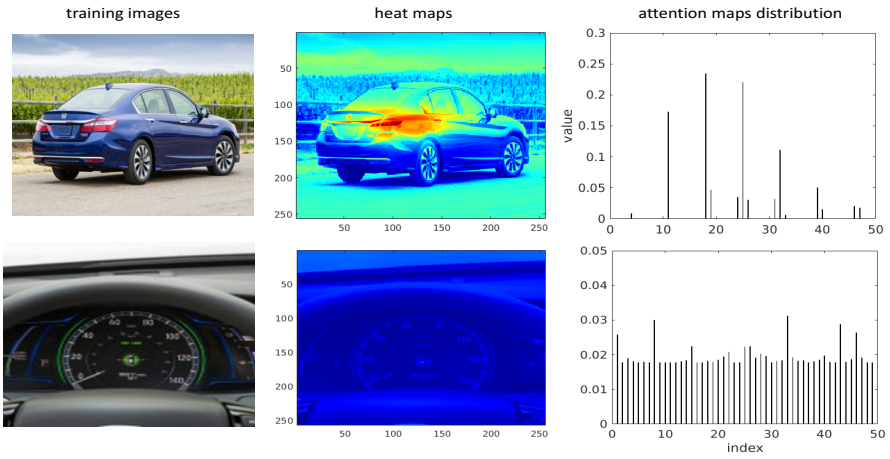


Figure 5.2: This figure illustrates the effectiveness of the group-wise attention model used in the proposed method. The left column shows the original training images. The middle column is the images plus its corresponding attention heat maps. The right column shows the distribution of the attention maps. The upper row relates to the correctly labeled sample and the bottom row corresponds to the mislabeled sample. We can see that for the correctly labeled sample, the normalized attention model only focus on the discriminative local parts and the score distribution is sparse. In contract, for the mislabeled sample, the normalized attention model fails to concentrate on any local regions and the score distribution is dense.

network training. Thus in practice, we choose K as a small value (2 to 5). We have conducted an experimental study on the impact of K with respect to different level of labeling noise at Section 5.4.4.

5.3.2 Attention

5.3.2.1 Attention formulation

After random grouping, each instance is now represented as an array of activations. These activations come from both correctly labeled images and mislabeled images. Although containing activations from the correct region of interest, many of the activations are noisy signals and will negatively impact the learning process. To mitigate this issue, we propose to use an attention model to focus processing only on the attended activations. Let $\mathbf{x}_{ijk}^{n} \in \mathbb{R}^{c}$ denote the last convolutional layer activations from the k-th image of the n-th instance at the spatial location (i, j), where $i = 1, 2, ..., d$

and $j = 1, 2, ..., d$ are the coordinates of the feature map and d is the height or width of the feature map.

The unnormalized attention score $s_{ijk}^n \in \mathbb{R}$ can be formulated as

$$s_{ijk}^n = f(\mathbf{w}^{\mathrm{T}} \mathbf{x}_{ijk}^n + b), \tag{5.2}$$

where $\mathbf{w} \in \mathbb{R}^c$, $b \in \mathbb{R}^1$ denote the weight and bias of the attention detector respectively, which are parts of the model parameters and will be learned in an end-to-end manner. $f(\cdot)$ is the softplus function $f(x) = \ln(1 + \exp(x))$. Since we are only concerned with the relative importance of the local features within an image, we propose to normalize the attention scores to $[0, 1]$ for aggregating the local features:

$$a_{ijk}^n = \frac{s_{ijk}^n + \varepsilon}{\sum_i \sum_j (s_{ijk}^n + \varepsilon)}, \tag{5.3}$$

where a_{ijk}^n is the normalized attention score, ε is a small constant and quite important to make the distribution reasonable.

If the element s_{ijk}^n is low but there is no ε, then the corresponding a_{ijk}^n can be large even though s_{ijk}^n is small. The constant ε can solve this problem effectively. If it is properly set, a small s_{ijk}^n (approaching zero) will result in $a_{ijk}^n = \frac{1}{d^2}$. In our work, we set it to 0.1.

After obtaining the normalized attention scores, we can get the attended feature representation by applying a_{ijk}^n to $\mathbf{x}_{ijk}^n$ as follows:

$$\widehat{\mathbf{x}}_{ijk}^n = a_{ijk}^n \odot \mathbf{x}_{ijk}^n, \tag{5.4}$$

where $\odot$ is the element-wise multiplication, $\widehat{\mathbf{x}}_{ijk}^n$ is the attended feature representation.

Then the representation of a grouped training instance can be obtained by a global average pooling over all the feature dimensions except for the channel-wise dimen-

sion:

$$\mathbf{h}_n = \frac{1}{d^2 k} \sum_i \sum_j \sum_k \widehat{\mathbf{x}}_{ijk}^n, \tag{5.5}$$

where $\mathbf{h}_n \in \mathbb{R}^c$ is the group-level representation of the n-th training instance.

Then we apply a linear classifier layer to predict the class label of each grouped instance and use the multi-class cross-entropy loss to train the network:

$$L_{class} = -\sum_n y_n \log\left(\frac{\exp(\mathbf{F}_n)}{\sum_n \exp(\mathbf{F}_n)}\right) \tag{5.6}$$

where $\mathbf{F}_n$ and y_n are the last linear classification layer and the class label for the n-th training instance, respectively.

5.3.2.2 Attention module regularization

Ideally, for the correctly labeled image, the attention scores should have large values on one or few image regions; for the mislabeled image, none of the image regions should correspond to large attention values. In the above framework, we expect this situation can happen after the end-to-end training of the network. In this section, we devise a regularization term to further encourage this property. To apply this regularization, we assume that a set of negative class images belonging to none of to-be-learned image categories is available. Then we can apply the attention detector on those negative class images and require that the obtained normalized attention values are as small as possible since those images do not contain the object-of-interest. Define $u_{ijk}^n = \mathbf{w}^{\mathrm{T}} \mathbf{x}_{ijk}^n + b$ to be the linear attention scores for the sample $\mathbf{x}_{ijk}^n$; then the above requirement is equivalent to expecting $\max_{ijk} u_{ijk}^n < 0$. On the other hand, for a grouped training instance generated from each class, we expect that the attention detector identifies at least one relevant region and this leads to the objective $\max_{ijk} u_{ijk}^n > 0$. In this chapter, we propose to use the following objective function to

impose the aforementioned two requirements:

$$R(\mathbf{w}, b) = \sum_{n} \max(0,\ 1 - \delta_n \max_{ijk}(u^n_{ijk})) \tag{5.7}$$

where $\delta_n = \{1, -1\}$ indicates whether the instance is sampled from the classes of object-of-interest or from the negative class. We then use the weighted sum of L_{class} and R as the final objective function:

$$L = L_{class} + \lambda R. \tag{5.8}$$

The effect of the attention module is illustrated in Figure 6.3. The input is an instance including a correctly labeled car sample and a mislabeled noise sample. We can observe that for the correctly labeled sample, the normalized attention scores are pushed high at the region-of-interest, which corresponds to the back of the car in the example. In contrast, for the mislabeled sample, the normalized attention scores are all pushed approaching zero, resulting in no parts to be concentrated on for the attention model. In terms of this observation, we can explore that the attention model can not only filter out the contextual features of the mislabeled samples in the training instance, but also help detect the discriminative parts of the correctly labeled samples.

5.4 Experiments

In this section, we test our weakly-supervised learning framework on two datasets collected from the Web. One is a fine-grained dataset and the other one is a conventional classification dataset. The training data for both tasks are obtained via search results freely available from Google image search, using all returned images as training data. It's worth noticing that fine-grained classification is quite challenging because categories can only be discriminated by subtle and local differences.

Figure 5.3: Examples of the image re-ranking performance on one sampled car category ("cadillac"). The red crosses indicate the images that are classified incorrectly. The images are sorted according to the rank of the classification scores in descending order. The images in the green rectangle and red rectangle are correctly labeled samples and mislabeled samples, respectively. The noise level is 0.4.

5.4.1 Datasets

WebCars: We collect a large-scale fine-grained car dataset from the internet, named WebCars, using the categories of the clean CompCars dataset [Yang et al., 2015]. We treat the car model names as the query keywords and automatically retrieve images for all the 431 fine-grained categories. We collect 213,072 noisy Web images in total and still use the test set of the original clean dataset for testing. We sample a few categories from WebCars and manually annotate the ground-truth labels, noting in the process that approximately 30% of images are outliers. We further collect 10,000 images that doesn't belong to the training categories as the negative class.

Web data + ImageNet: We randomly sample 100 classes used in ImageNet and use the category names for collecting a noisy Web image dataset. All the images are automatically downloaded and the ones that appear in the original ImageNet dataset are manually removed. This dataset contains 61,639 images in total. The noise gradually increases from the highly ranked images to the latter samples. We estimate the percentage of mislabeled samples is approximately 20 %. We also collect 5,000 negative class Web images.

5.4.2 Implementation details

We use Theano [Bastien et al., 2012] for our experiments. We use the pretrained VGG-16 model trained on the ImageNet dataset [Russakovsky et al., 2015a] to initialize the convolutional layers of our framework. The learning rate is set to 0.001 initially, and divided by 10 after 5 epoches. The regularizer λ is set to 0.1. Training samples are randomly grouped online.

To investigate the impact of the various elements in our end-to-end framework, we analyse the effects of the attention model, group-wise training approach and the attention regularization described in Section 5.3.2.2 independently.

1. "Average pooling without attention (AP)": We employ the average-pooling method as an important baseline here since it's commonly used for image classification on clean images without any noise-robust strategy. The average pooling structure simply replaces the two 4096 dimensions fully-connected layers in VGG-16 model with an average pooling layer, followed by a softmax layer for classification.

2. "Random grouping training without attention (RGT)": In this method, samples are randomly grouped during training, with the mean-pooling operation in Eq. 5.5 to get the instance-level representation.

3. "Average pooling with attention (AP+AT)": Based on AP, the attention model is embedded in the network to test its ability to localize discriminative feature regions.

4. "Random grouping training with attention (RGT+AT)": Attention is added to RGT.

5. "Average pooling with attention and regularizer (AP+AT+R)": We add the regularizer to AP+AT to evaluate its influence to cope with noisy labels.

6. "Random grouping training with attention and regularizer (RGT+AT+R)": We

test its performance on filtering out incorrectly labeled samples in each group as well as noisy local feature parts by adding the regularizer to RGT+AT.

5.4.3 Evaluation on the WebCars

We quantatively compare the methods described in Section 5.4.2 and report the results in Table 5.1. For RGT based methods, the group size is set to 2.

methods	accuracy
AP	66.86%
RGT	69.83%
AP+AT	73.64%
RGT+AT	76.58%
AP+AT+R	70.77 %
RGT+AT+R	**78.44**%

Table 5.1: Comparison of classification results on the Compcars test set.

Average pooling vs. Random grouping training

By comparing the results of AP and RGT, we can see that the group-wise training can effectively suppress the influence of noise due to the improved labeling accuracy at the instance level. For this reason, the model can always learn some useful information from the correctly labeled samples in each group. In contrast, for training at the image level with no attention, the noisy labels will give networks misleading information that will harm the learning process.

Attention vs. without attention

For AP+AT and RGT+AT, the accuracy all improves by a large margin compared to AP and RGT respectively, which proves the effectiveness of the attention model employed. The attention model filters out uninformative parts of the feature maps for each sample and only let the useful parts flow through the latter network for classification. In this way, it works like a gate that can prevent the noisy regions of the feature representation from misleading the classifiers. A similar strategy is found effective on clean images for multi-label image classification [Zhao et al., 2016].

With vs. without regularizer

An interesting phenomenon we observe is that the accuracy for AP+AT drops

significantly when using the noise regularizer, AP+AT+R. The reason is that the noise presents in both classes of object-of-interest and negative class, and consequently the image-level learning strategy confuses the network with how to classify the noise. But this confusion doesn't exist in the group-level training approach, since very few training instances have incorrect labels after random grouping. The reasons for adding noise regularizer is helpful for group-wise training are two-fold: First, the hinge loss regularizer forces the attention map not to concentrate on any feature regions of mislabeled samples, which results in a much cleaner group-level feature representation; Second, it helps the classifiers to distinguish the correctly labeled samples from the noise [Girshick et al., 2014]. It's worth noticing that compared to utilizing clean images as constraint [Xiao et al., 2015], the negative samples are much easier to collect.

We consider two types of label noise defined in [Krause et al., 2016], which are called *cross-domain* noise and *cross-category* noise. The cross-domain noise is defined to be the portion of images that are not of any category in the fine-grained domain, *i.e.* for cars, these images don't contain a car. In contrast, the cross-category noise is the mislabeled images within a fine-grained domain, *i.e.* a car example with the wrong model label.

We also provide qualitative examples in Figure 5.5. We see that the attention model mostly focuses on the discriminative parts in the front of or at the end of the cars. For some challenging examples, the correctly labeled car appears simultaneously with the cross-domain noise or cross-category noise in the same image. In this case, the attention model still successfully localizes to the correct parts. For the mislabeled samples, there's no object-of-interest to be concentrated on.

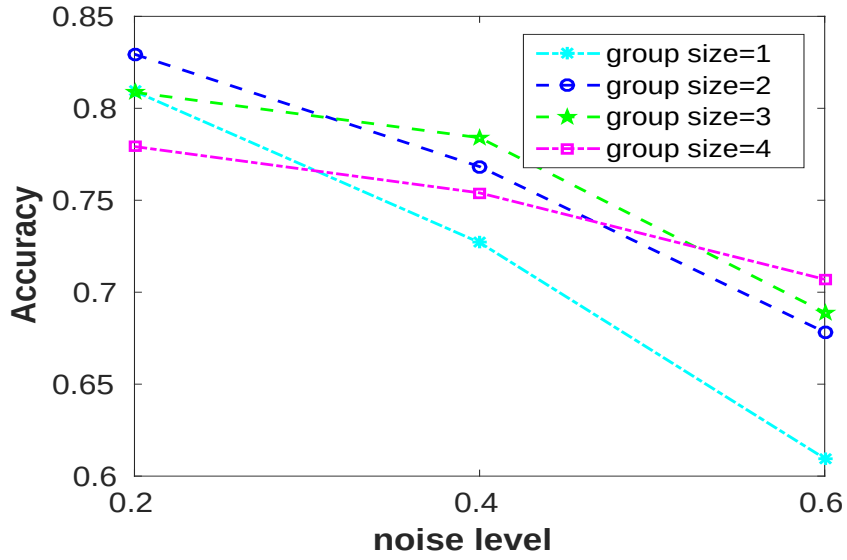


Figure 5.4: The classifcation accuracy under different group sizes of the proposed method.

5.4.4 Analysis of group size

In this section, we conduct a toy experiment to investigate the impact of the group size on our method (RGT+AT+R)[1]. We randomly sample 100 car categories of the Compcars dataset and deliberately pollute the clean training data by adding cross-category noise and cross-domain noise in a proportion of 1:1. The total number of training images doesn't change. We then gradually increase the noise level from 0.2 to 0.6 and report the classification accuracy on the test set of Compcars using different group sizes. The results are shown in Figure 5.4. From Figure 5.4, we could make the following observations: (1) using group size ≥ 2 makes the network training more robust to noise. As can be seen, when the dataset contains a substantial amount noise label e.g. noise level = 0.6, the performance gap between group size = 1 and group size ≥ 2 can be larger than 10%. (2) the optimal group size changes with the noise level. For example, when the noise level = 0.2, the optimal group size is 2 but when the noise level = 0.6, the optimal group size becomes 4. This observation could

[1]When group size equals 1, the method is equivalent to AP + AT + R. We empirically find adding the regularization term in this case will lead to inferior performance so we do not use the regularization term when group size equals 1.

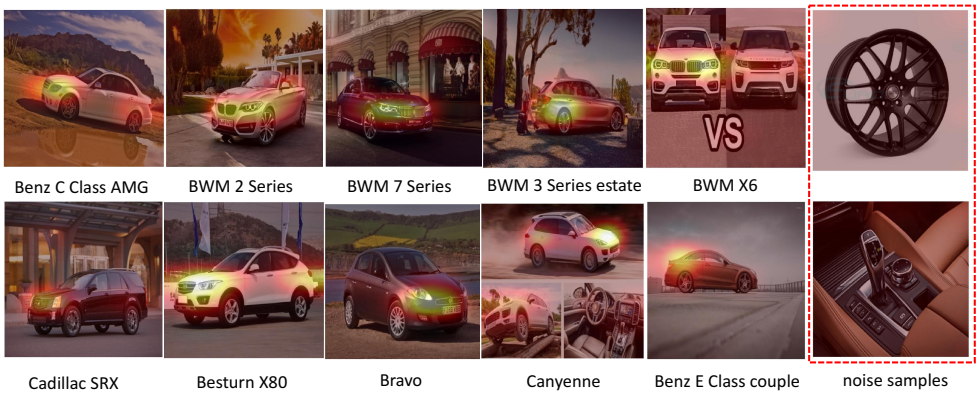


Figure 5.5: Examples of the attention maps using the large-scale noisy fine-grained dataset described in Section 5.4.1. The brighter the region, the higher the attention scores. The examples in the red dotted box are mislabeled samples on the Web.

Figure 5.6: Examples of where attention maps for the collected Web data with respect to ImageNet described in Section 5.4.1. The brighter the region, the higher the attention scores. The examples in the red dotted box are mislabeled on the Web.

be partially explained by the analysis in section 5.3.1, that is, the larger group size reduces the chance of having an incorrect label at the group-level. (3) Finally, we observe that larger k does not always lead to better performance. As also mentioned in section 5.3.1, we speculate that this is because having a larger group will reduce the independency of grouped instance. For example, when having a larger k, the chance of two groups sharing one common image will grow significantly.

5.4.5 Web Images re-ranking

To inspect whether the proposed method utilize the information from the correctly labeled data for training while ignoring the mislabeled ones, we now propose to re-rank the noisy training data used in Section 5.4.4 according to their classification scores. The ideal case is that the highly ranked images are all correctly labeled ones while the low-ranking samples are mislabeled ones on the Web. We compare three methods here, including AP, AP+AT as well as RGT+AT+R using different group sizes. The ground truth labels for correctly labeled images and mislabeled images are set to $+1$ and -1, respectively. Correctly labeled images are ranked high in the ground truth labels. Based on the learned models in Section 5.4.4, we first obtain the classification score for each training sample and rank the images in descending order based on their corresponding classification scores to get the predicted labels in each category. We then calculate the mean average precision (MAP) under different noise levels and group sizes. The mean average precision is obtained by averaging the precisions calculated at the total number of samples in different categories.

methods \ noise level	20 %	40 %	60 %
AP	93.72	85.08	74.42
AP+AT	96.71	92.84	90.56
RGT+AT+R, group size=2	**98.12**	95.81	91.00
RGT+AT+R, group size=3	97.71	**95.93**	91.04
RGT+AT+R, group size=4	97.95	95.33	**91.98**

Table 5.2: Comparison of mean average precisions % using several methods under different noise levels.

From the table, we can see that for direct average pooling, the precision drops dramatically as the noise level increases. On the contrary, simply adding attention model only, the precision improves considerably especially when the noise level is high enough. For example, at the noise level 60 %, the precision gap is more than 15 %. This result proves that selecting discriminative regions for each sample can effectively prevent noisy parts from impacting the final classification. By incorpo-

rating the group-wise training strategy, the performance further improves. This can be attributed to the highly accurate group-level labels used and the attention model for blocking the local features of mislabeled samples to generate the group-level representation. Overall, the proposed method is stable and performs well at different noise levels.

We also randomly select a car category and qualitatively evaluate the re-ranking performance at the noise level 0.4 (see Figure 5.3). The images are ranked in descending order based on their classification scores. We can see that only a pair of images are ranked incorrectly among the samples. From the results, we can expect that our method can further be used to assist collecting clean datasets or active learning.

5.4.6 Evaluation on CIFAR-10 with Synthetic Noises

We also conduct synthetic experiments on CIFAR-10 following the setting of [Xiao et al., 2015; Sukhbaatar and Fergus, 2015] and report the test accuracies under different noise levels in Table 5.3. As seen, the proposed method is more robust to label noise.

methods \ noise level	30 %	40 %	50 %
Caffe's CIFAR10-quick	65.57%	62.38%	57.36%
[Sukhbaatar and Fergus, 2015]	69.73%	66.66%	63.39%
[Xiao et al., 2015]	69.81%	66.76%	63.00%
RGT+AT+R, group size=2	**74.88** %	70.33%	65.87%
RGT+AT+R, group size=3	71.76 %	**72.25**%	**67.15**%
RGT+AT+R, group size=4	70.23 %	70.74%	66.98%

Table 5.3: Accuracies on CIFAR-10 with synthetic label noises.

5.4.7 Evaluation on Web Images + ImageNet

Apart from the challenging fine-grained classification task, the proposed method can also be generalized to a conventional classification task. We trained models from

scratch using the noisy Web data with respect to ImageNet described in Section 5.4.1 and test the performance on the ILSVRC2012 validation set.

methods	accuracy
AP	58.81%
AP+AT	67.68%
RGT+AT+R, group size=2	**71.24**%
RGT+AT+R, group size=3	68.89%
RGT+AT+R, group size=4	66.23%

Table 5.4: Comparison of classification results on ILSVRC2012 test set.

From the results we can see that for the conventional image classification task with Web data, the proposed method still works much better than the directly average pooling baseline. By only applying the attention model on each sample to select discriminative feature regions for classification, the result improves by $\sim 9\%$. By randomly generating groups online using reasonable group size and incorporating the regularizer, we get the best performance at the optimal group size 2, which confirms the conclusions in Section 5.4.3 and Section 5.4.4.

We visualize some examples with their attention maps in Figure 5.6 using the best performed method RGT+AP+R with group size 2. The attention model attempts to localize the most discriminative parts for correctly labeled samples to push them far from the decision boundary. Samples in the red bounding box are mislabeled on the Web and the attention model finds no parts to concentrate on.

5.5 Summary

In this chapter, we propose a weakly-supervised framework to learn visual representations from massive Web data with minor human supervision. The proposed method can handle label noise effectively by two unified strategies. By randomly stacking training images into groups, the accuracy of the group-level labels improves. The attention model embedded further localizes the discriminative regions corresponding to correctly labeled samples across the combined feature maps for

classification. The efficacy of our methods have been demonstrated by the extensive experiments.

Chapter 6

Towards Context-aware Interaction Recognition for Visual Relationship Detection

Statement of Authorship

Title of Paper	Towards context-aware Interaction Recognition for Visual Relationship Detection
Publication Status	☒ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Published at the International Conference on Computer Vision (ICCV), 2017.

Principal Author

Name of Principal Author (Candidate)	Bohan Zhuang
Contribution to the Paper	Wrote the paper and completed the experiments.
Overall percentage (%)	90%
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.
Signature	Date 7/12/2017

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Lingqiao Liu
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/2017

Name of Co-Author	Chunhua Shen
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/2017

Please cut and paste additional co-author panels ~~here as~~ required.

Name of Co-Author	Ian Reid		
Contribution to the Paper	Helped design the method and modified the paper.		
Signature		Date	7/12/17

6.1 Overview

Recognizing how objects interact with each other is a crucial task in visual recognition. If we define the context of the interaction to be the objects involved, then most current methods can be categorized as either: (i) training a single classifier on the combination of the interaction and its context; or (ii) aiming to recognize the interaction independently of its explicit context. Both methods suffer limitations: the former scales poorly with the number of combinations and fails to generalize to unseen combinations, while the latter often leads to poor interaction recognition performance due to the difficulty of designing a context-independent interaction classifier.

To mitigate those drawbacks, this chapter proposes an alternative, context-aware interaction recognition framework. The key to our method is to explicitly construct an interaction classifier which combines the context, and the interaction. The context is encoded via word2vec into a semantic space, and is used to derive a classification result for the interaction. The proposed method still builds one classifier for one interaction (as per type (ii) above), but the classifier built is adaptive to context via weights which are context dependent. The benefit of using the semantic space is that it naturally leads to zero-shot generalizations in which semantically similar contexts (subject-object pairs) can be recognized as suitable contexts for an interaction, even if they were not observed in the training set. Our method also scales with the number of interaction-context pairs since our model parameters do not increase with the number of interactions. Thus our method avoids the limitation of both approaches. We demonstrate experimentally that the proposed framework leads to improved performance for all investigated interaction representations and datasets.

6.2 Introduction

Object interaction recognition is a fundamental problem in computer vision and it can serve as a critical component for solving many visual recognition problems such

as action recognition [Mallya and Lazebnik, 2016; Ramanathan et al., 2015; Wang et al., 2015b; Bilen et al., 2016; Zhang et al., 2016a], visual phrase recognition [Hu et al., 2017; Rohrbach et al., 2016; Li et al., 2017a], sentence to image retrieval [Ma et al., 2015; Karpathy and Fei-Fei, 2015] and visual question answering [Wu et al., 2016c; Lu et al., 2017; Wu et al., 2016b]. Unlike object recognition in which the object appearance and its class label have a clear association, the interaction patterns, e.g., "eating", "playing", "stand on", usually have a vague connection to visual appearance. This phenomenon is largely caused by the same interaction being involved with different objects as its context, i.e. the subject and object of an interaction type. For example, "cow eating grass" and "people eating bread" can be visually dissimilar although both of them have the same interaction type "eating". Thus the subject and object associated with the interaction – also known as the *context* of the interaction – could play an important role in interaction recognition.

In existing literature, there are two ways to model the interaction and its context. The first one treats the combination of interaction and its context as a single class. For example, in this approach, two classifiers will be built to classify "cow eating grass" and "people eating bread." To recognize the interaction "eating", images that are classified as either "cow eating grass" or "people eating bread" will be considered as having interaction "eating". This treatment has been widely used in defining action (interaction) classes in many action (interaction) recognition benchmarks [Mallya and Lazebnik, 2016; Ramanathan et al., 2015; Wang et al., 2015b; Bilen et al., 2016; Zhang et al., 2016a]. This approach, however, suffers from poor scalability and generalization ability. The number of possible combinations of the interaction and its context can be huge, and thus it is very inefficient to collect training images for each combination. Also, this method fails to generalize to an unseen combination even if both its interaction type and context are seen in the training set.

To handle these drawbacks, another way is to model the interaction and the context separately [Lu et al., 2016; Desai et al., 2011; Gupta and Davis, 2008; Sadeghi

et al., 2015]. In this case, the interaction is classified independently of its context, which can lead to poor recognition performance due to the difficulty of associating the interaction with certain visual appearance in the absence of context information. To overcome the imperfection of interaction classification, some recent works employ techniques such as language priors [Lu et al., 2016] or structural learning [Li et al., 2017a; Liang et al., 2017a; Li et al., 2017b] to avoid generating an unreasonable combination of interaction and context. However, the context-independent interaction classifier is still used as a building block, and this prevents the system from gaining more accurate recognition from visual cues.

The solution proposed in this chapter aims to overcome the drawbacks of both methods. To avoid the explosion of the number of classes, we still separate the classification of the interaction and the context into two stages. However, different to the second method, the interaction classifier in our method is designed to be adaptive to its context. In other words, for the same interaction, different contexts will result in different classifiers and our method will encourage interactions with similar contexts to have similar classifiers. By doing so, we can achieve context-aware interaction classification while avoiding treating each combination of context and interaction as a single class. Based on this framework, we investigate various feature representations to characterize the interaction pattern. We show that our framework can lead to performance improvements for all the investigated feature representations. Moreover, we augment the proposed framework with an attention mechanism, which leads to further improvements and yields our best performing recognition model. Through extensive experiments, we demonstrate that the proposed methods achieve superior performance over competing methods.

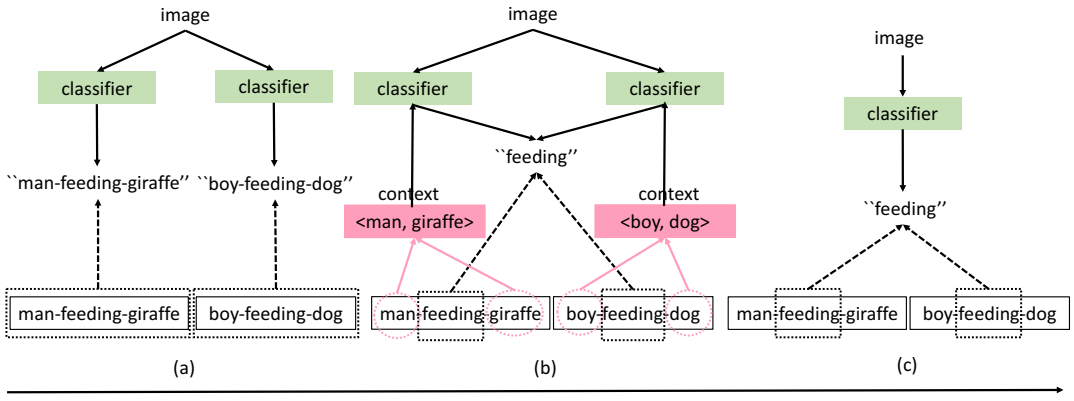


Figure 6.1: Comparison of two baseline interaction recognition methods and the proposed approach. The two baseline methods take two extremes. For one extreme, (a) treats the combination of the interaction and its context as a single class. For another extreme, (c) classifies the interaction separately from its context. Our method (b) lies somewhere between (a) and (c). We still build one classifier for each interaction but the classifier parameter is also adaptive to the context of the interaction, as shown in the example in (b).

6.3 Methods

6.3.1 Context-aware interaction classification framework

In general, an interaction and its context can be expressed as a triplet $\langle O1\text{-}P\text{-}O2 \rangle$, where P denotes the interaction, and $O1$ and $O2$ denote its subject and object respectively. In our study, we assume the interaction context ($O1$,$O2$) has been detected by a detector (i.e. we are given bounding boxes and lables for both subject $O1$ and object $O2$) and the task we are addressing is to classify their interaction type P. To recognize the interaction, existing works take two extremes in designing the classifier. One is to directly build a classifier for each P and assume that the same classifier applies to P with different context. Another takes the combination of $\langle O1\text{-}P\text{-}O2 \rangle$ as a single class and build a classifier for each combination. As discussed in the introduction section, the former does not fully leverage the contextual information for interaction recognition while the latter suffers from the scalability and generalization issues. Our proposed method lies between those two extremes. Specifically, we still allocate one classifier for each interaction type, however we make the classifier parameters adap-

tive to the context of the interaction. In other words, the classifier is a function of the context. The schematic illustration of this idea is shown in Figure 6.1.

Formally, we assume that the interaction classifier takes a linear classifier form $y_p = \mathbf{w}_p^\top \phi(I), \ \ \mathbf{w}_p \in \mathbb{R}^d$, where y_p is the classification score for the p-th interaction and $\phi(I)$ is the feature representation extracted from the input image. The classifier parameters for the p-th interaction $\mathbf{w}_p$ are a function of $(O1, O2)$, that is, the context of the p-th interaction. It is designed as the summation of the following two terms:

$$\mathbf{w}_p(O1, O2) = \bar{\mathbf{w}}_p + r_p(O1, O2), \tag{6.1}$$

where the first term $\bar{\mathbf{w}}_p$ is independent of the context; it plays a role which is similar to the traditional context-independent interaction classifier. The second term $r_p(O1, O2)$ can be viewed as an auxiliary classifier generated from the information of context $(O1, O2)$. Note that the summation of two classifiers has been widely used in transfer learning [Patricia and Caputo, 2014; Arnold et al., 2007; Do and Ng, 2005] and multi-task learning [Evgeniou and Pontil, 2004; Parameswaran and Weinberger, 2010], e.g., one term corresponds to the classifier learned in the target domain and another corresponds to the classifier learned in the source domain.

Intuitively, for two interaction-context combinations, if both of them share the same interaction and their contexts are similar, the interaction in those combinations tends to be associated with similar visual appearance. For example, $\langle$*boy, playing, football*$\rangle$ and $\langle$*man, playing, soccer*$\rangle$ share similar context, so the interaction "playing" should suggest similar visual appearance for these two combinations. This inspires us to design $\mathbf{w}_p(O1, O2)$ to allow semantically similar contexts to generate similar interaction classifiers, as demonstrated in Figure 6.2. To realize this idea, we first represent the object and subject through their word2vec embedding which maps semantically similar words into similar vectors and then generate the auxiliary classifier r_p by

concatenating their embeddings. Formally, r_p is designed as:

$$r_p(O1, O2) = \mathbf{V}_p f(\mathbf{Q}E(O1, O2)), \tag{6.2}$$

where $E(O1, O2) \in \mathbb{R}^{2e}$ is the concatenation of the e-dimensional word2vec embeddings of $(O1, O2)$, and $\mathbf{Q} \in \mathbb{R}^{m \times 2e}$ is a projection matrix to project $E(O1, O2)$ to a low-dimensional (e.g. 20) semantic embedding space. $f(\cdot)$ is the RELU function and $\mathbf{V_p}$ transforms the context embedding to the auxiliary classifier. Note that $\mathbf{V}_p$ and $\bar{\mathbf{w}}_p$ in Eq. (6.1) are distinct per interaction type p while the projection matrix $\mathbf{Q}$ is shared across all interactions. All of these parameters are learnt at training time.

Remark: Many recent works [Liang et al., 2017a; Li et al., 2017a; Zhang et al., 2017a; Plummer et al., 2016] on visual relationship detection takes a structural learning alike formulation to simultaneously predict $O1, O2$ and P. The unary term used in their framework is still a context-independent classifier and such choice may lead to poor recognition accuracy in identifying interaction from the visual cues. To improve these techniques, one could replace their unary terms with our context-aware interaction recognition module. On the other hand, their simultaneous prediction framework could also benefit our method in achieving better visual relationship performance. Since our focus is to study the interaction part, we do not pursue this direction in this chapter and leave it for future work.

6.3.2 Feature representations for interactions recognition

One remaining issue in implementing the framework in Eq. (6.1) is the design of $\phi(I)$, that is, the feature representation of the interaction. It is clear that the choice of the feature representation can have significant impact on the interaction prediction performance. In this section, we investigate two types of feature representations to characterize the interaction. We evaluate these feature representations in Sec. 6.4.1.1.

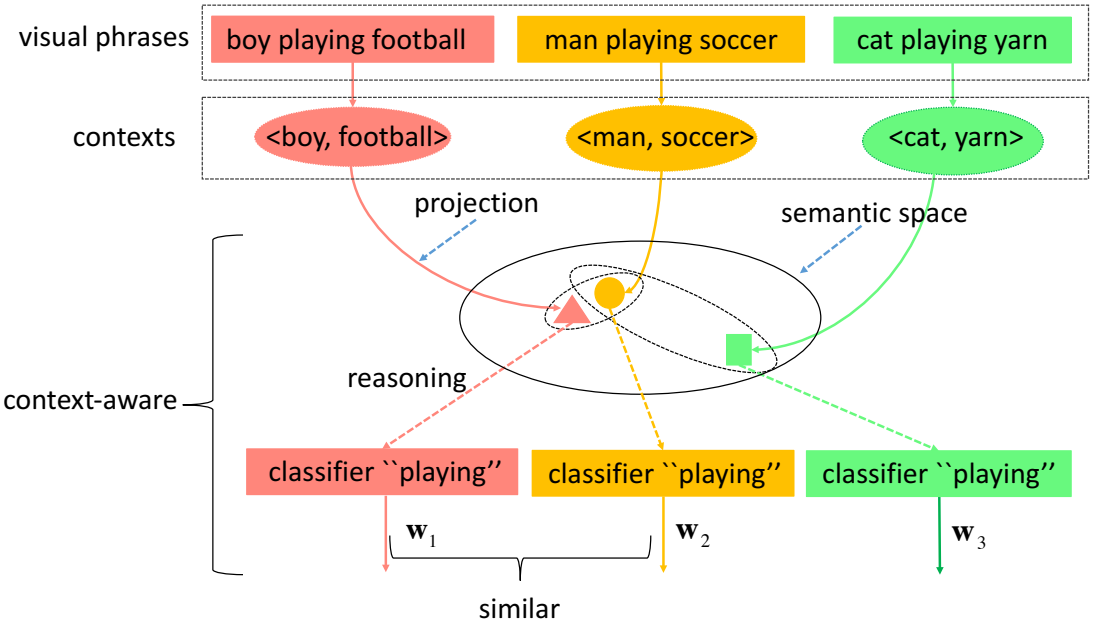


Figure 6.2: An example of the proposed context-aware model. The same interaction "playing" is associated with various contexts. The contexts of the first two phrases are semantically similar, resulting in two similar context-aware classifiers. Since the last two contexts are far away from each other in the semantic space, their corresponding context-aware classifiers may not similar despite sharing the same label. In this way, we explicitly consider the visual appearance variations introduced by changing context, thus more accurate and generalizable interaction classifiers can be learned.

6.3.2.1 Spatial feature representation

Our method assumes that the context has been detected and therefore the interaction between the subject and the object could be characterized by the spatial features of the detection bounding boxes. These kind of features have been previously employed [Hu et al., 2017; Plummer et al., 2016; Zhang et al., 2017a] to recognize the visual relationship of objects. In our study, we use both the spatial features from each bounding box and the spatial features from their mutual relationship. Formally, let (x, y, w, h) and (x', y', w', h') be the bounding box coordinates of the *subject* and *object*, respectively. Given the bounding boxes, the spatial feature for a single box is a 5-dimentional vector represented as $[\frac{x}{W_I}, \frac{y}{H_I}, \frac{x+w}{W_I}, \frac{y+h}{H_I}, \frac{S_b}{S_I}]$, where S_b and S_I are the areas of region b and image I, W_I and H_I are the width and height of the image I. And the

pairwise spatial vector is denoted as $[\frac{x-x'}{w'}, \frac{y-y'}{h'}, \log\frac{w}{w'}, \log\frac{h}{h'}]$. We concatenate them together to get a 14-dimentional feature representation (using both subject and object bounding boxes). Then the spatial feature directly passes through the context-aware classifier defined in Eq. (6.1) for the interaction classification.

6.3.2.2 Appearance feature representation

Besides spatial features, we can also use appearance features, e.g. the activations of a deep neural network to depict the interaction. In our study, we first crop the union region of the subject and object bounding boxes, and rescale the region to $224 \times 224 \times 3$ as the input of a VGG-16 [Simonyan and Zisserman, 2015] CNN. We then apply the mean-pooling to the activations of the $conv5\_3$ layer as our feature representation $\phi(I)$. This feature is then fed into our context-aware interaction classifier in Eq. (6.1). To improve the performance, we treat the context-aware interaction classifier as a newly added layer and fine-tune this layer with the VGG-16 net in an end-to-end fashion.

6.3.3 Improving appearance representation with attention and context-aware attention

The discriminative visual cues for interaction recognition may only appear in a small region of the input image or the image region. For example, to see if "man riding bike" occurs, one may need to focus on the region near human feet and bike pedal. This consideration motivates us to use attention module to encourage the network "focus on" discriminative regions. Specially, we can replace the mean-pooling layer in Sec. 6.3.2.2 with an attention-pooling layer.

Formally, let $\mathbf{h}_{ij} \in R^c$ denote the last convolutional layer activations at the spatial location (i, j), where $i = 1, 2, ..., M$ and $j = 1, 2, ..., N$ are the coordinates of the feature map and M, N are the height and width of the feature map respectively, c is the number of channels. The attention pooling layer pools the convolutional layer

activations into a c-dimensional vector through:

$$\begin{aligned} \bar{a}(\mathbf{h}_{ij}) &= \frac{a(\mathbf{h}_{ij})+\varepsilon}{\sum_i \sum_j (a(\mathbf{h}_{ij})+\varepsilon)}, \\ \widetilde{\mathbf{h}} &= \frac{1}{MN} \sum_{ij} \bar{a}(\mathbf{h}_{ij}) \mathbf{h}_{ij}, \end{aligned} \tag{6.3}$$

where $a(\mathbf{h}_{ij})$ is the attention generation function which produces an attention value for each location (i, j). The attention value is then normalized (ε is a small constant) and used as a weighting factor to pool the convolutional activations $\mathbf{h}_{ij}$. We consider two designs of $a(\mathbf{h}_{ij})$.

Direct attention: The first attention generation function is simply designed as $a(\mathbf{h}_{ij}) = f(\mathbf{w}_{att}^{\top} \mathbf{h}_{ij} + b)$, where $\mathbf{w}_{att}$ and b are the weight and bias of the attention model.

Context-aware attention In the above attention generation function, the attention value is solely determined by $\mathbf{h}_{ij}$. Intuitively, however, it makes sense that different attention is required for different classification tasks. For example, to examine "man riding bike" and examine "man playing football", different regions-of-interest should be focused on. We therefore propose to use a context-aware attention generator; i.e. we design $\mathbf{w}_{att}$ as a function of $(P, O1, O2)$. We can follow the framework in Eq. (6.1) to calculate:

$$\mathbf{w}_{att}(P, O1, O2) = \bar{\mathbf{w}}_p^{\mathbf{a}} + \mathbf{V}_p^a f(\mathbf{Q} E(O1, O2)), \tag{6.4}$$

where $\bar{\mathbf{w}}_p^{\mathbf{a}}$ is the attention weight for the p-th interaction independent of its context and $\mathbf{V}_p^a$ transforms the semantic embedding of the context to the auxiliary attention weight for the p-th interaction. Note that in this case $\mathbf{w}_{att}$ depends on the interaction class P and therefore different attention-pooling vectors $\widetilde{\mathbf{h}}_p$ will be generated for different P. $\widetilde{\mathbf{h}}_p$ will be then sent to the context-aware classifier for interaction P to obtain the decision value for P and the class that produces the maximal decision value will be considered as the recognized interaction. This structure is illustrated in Figure 6.3.

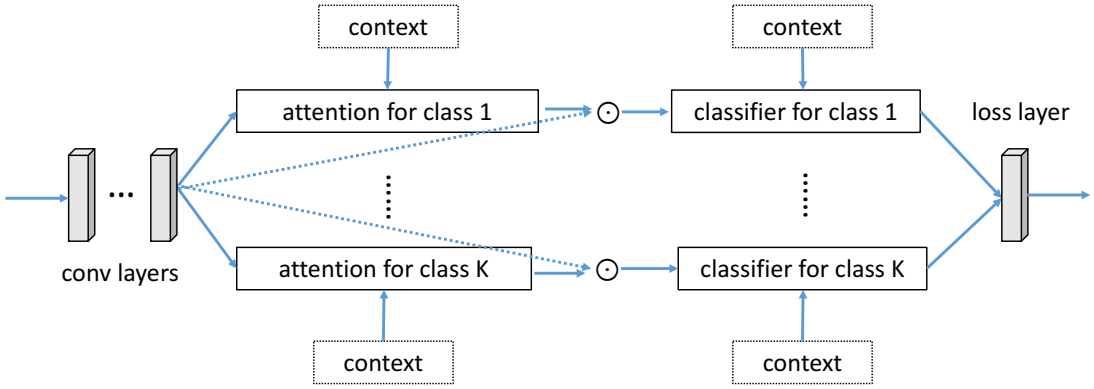


Figure 6.3: Detailed illustration of the context-aware attention model. For each interaction class, there is a corresponding attention model imposed on the feature map to select the interaction-specific discriminative feature regions. Different attention-pooling vectors will be generated for different interaction classes. The generated pooling vector will be then sent to the corresponding context-aware classifier to obtain the decision value.

6.3.4 Implementation details

For all the above methods, we use the standard multi-class cross-entropy loss to train the models. The Adam algorithm [Kingma and Ba, 2014] is applied as the optimization method. The methods that use appearance features involve convolutional layers from the standard VGG-16 network together with some newly added layers. For the former we initialize those layers with the parameters pretrained on ImageNet [Russakovsky et al., 2015b] and for the latter we randomly initialize the parameters. We set the learning rate to 0.001 and 0.0001 for the new layers and VGG-16 layers respectively.

6.4 Experiments

To investigate the performance of the proposed methods, we analyse the effects of the context-aware interaction classifier, the attention models and various feature representations. Eight methods are implemented and compared:

1. "**Baseline1-app**": We directly fine-tune the VGG-16 model to classify the in-

teraction categories. Inputs are the union of subject and object boxes. This baseline models the interaction and its context separately, which corresponds to the approach described in Figure 6.1 (c).

2. "**Baseline1-spatial**": We directly train a linear classifier to classify the spatial features described in Sec. 6.3.2.1 into multiple interaction categories.

3. "**Baseline2-app**": We treat the combination of the interaction and its context as a single class and fine-tune the VGG-16 model for classification. This corresponds to using appearance feature to implement the method in Figure 6.1 (a).

4. "**Baseline2-spatial**": Similar to "Baseline2-app". We train a linear classifier to classify the spatial features into the classes derived from the combination of the interaction and its context.

5. "**AP+C**": We apply the context-aware classifier to the appearance representation described in Sec. 6.3.2.2.

6. "**AP+C+AT**": The basic attention-pooling representation described in Sec. 6.3.3 with the classifier in **AP+C**.

7. "**AP+C+CAT**": The context-aware attention-pooling representation described in Sec. 6.3.3 with the classifier in **AP+C**.

8. "**Spatial+C**": We apply the context-aware classifier to the spatial features described in Sec. 6.3.2.1.

Besides those methods, we also compare the performance of our methods against those reported in the related literature. However, it should noted that these methods may use different feature representation, detectors or pre-training strategies.

6.4.1 Evaluation on the visual relationship dataset

We first conduct experiments on the Visual Relationship Detection (VRD) dataset [Lu et al., 2016]. This dataset is designed for evaluating the visual relationship ($\langle subject, predicate, object \rangle$) detection, where the "predicate" in those datasets is equivalent to the "interaction" in our chapter and we will use them interchangeably thereafter. It contains 4000 training and 1000 test images including 100 object classes and 70 predicates. In total, there are 37993 relationship instances with 6672 relationship types, out of which 1877 relationships occur only in the test set but not in the training set.

Following [Lu et al., 2016], we evaluate on three tasks: (1) For **predicate detection**, the input is an image and a set of ground-truth object bounding boxes. The task is to predict the possible interactions between pairs of objects. Since the interaction recognition is the main focus of this section, the performance of this task provides the most relevant indication of the quality of the proposed method. (2) In **phrase detection**, we aim to predict $\langle subject\text{-}predicate\text{-}object \rangle$ and localize the entire relationship in one bounding boxes. (3) For **relationship detection**, the task is to recognize $\langle subject\text{-}predicate\text{-}object \rangle$ and localize both subject and object bounding boxes. Both boxes should have at least 0.5 overlap with the ground truth bounding boxes in order to be regarded as a correct prediction. For the second and third tasks, we use the object detection results (both bounding boxes and corresponding detection scores) provided in [Lu et al., 2016]. This allows us to fairly compare the performance of the proposed interaction recognition framework without the influence of detection.

We use the Recall@100 and Recall@50 as our evaluation metric following [Lu et al., 2016]. Recall@x computes the fraction of times the correct relationship is calculated in the top x predictions, which are ranked by the product of the objectness confidence scores and the classification probabilities of the interactions. As discussed in [Lu et al., 2016], we do not use the mean average precision (mAP), which is a pessimistic evaluation metric because it cannot exhaustively annotate all possible relationships in an image.

6.4.1.1 Detection results comparison

In this section, we evaluate the performance of three detection tasks on the Visual Relationship Detection (VRD) benchmark dataset and provide the comprehensive analysis. We compare all the eight methods and the results in [Sadeghi and Farhadi, 2011; Lu et al., 2016]. The results are shown in Table 7.2. From it we can make the following observations:

The effect of context-aware modeling: To validate the main point in this chapter, we compare the proposed method against two context-interaction modeling baselines, i.e. baseline1-app, baseline2-app, baseline1-spatial and baseline2-spatial). By analysing the results, we can see that the proposed context-aware modeling methods (methods with "AP") achieves much better performance than the four baselines. The improvement achieved by use context-aware modeling is consistently observed for both spatial features and appearance features. This justifies that the context information is crucial for interaction prediction.

Various feature representations: We also quantitatively investigate the performance of the proposed context-aware framework under various feature types. As can be seen in Table 7.2, the appearance feature representation performs consistently better than the spatial feature representation, especially for the baseline2 setting. This may be because the visual feature representation has richer discriminative power than the 14-dimensional spatial feature. Also, with our context-aware recognition framework, we can significantly boost the performance of both features and interestly in this case the gap between two types of features is largely diminished, e.g. AP+C+CAT vs. Spatial+C.

The effect of attention models: We also investigate the impacts of the attention scheme employed in our model by comparing AP+C, AP+C+AT and AP+C+CAT. The best results are obtained by utilizing the context-aware attention model. This justifies our postulate that it is better to make the network attend on the discriminative regions of feature maps.

Method	**Predicate Det.**		Phrase Det.		Relationship Det.	
	R@100	R@50	R@100	R@50	R@100	R@50
Visual Phrase [Sadeghi and Farhadi, 2011]	1.91	0.97	0.07	0.04	-	-
Language Priors [Lu et al., 2016]	47.87	47.87	17.03	16.17	14.70	13.86
Baseline1-app	18.13	18.13	6.02	5.42	5.54	5.01
Baseline1-spatial	17.77	17.77	5.24	4.77	4.54	4.19
Baseline2-app	27.23	27.23	9.30	7.91	8.34	7.03
Baseline2-spatial	13.85	13.85	4.15	3.06	3.63	2.63
Spatial+C	51.17	51.17	17.61	15.46	15.43	13.51
AP+C	52.36	52.36	18.69	16.91	16.46	14.88
AP+C+AT	53.12	53.12	19.08	17.30	16.89	15.40
AP+C+CAT	**53.59**	**53.59**	**19.24**	**17.60**	**17.39**	**15.63**

Table 6.1: Evaluation of different methods on the visual relationship benchmark dataset. The results reported include visual phase detection (Phrase Det.), visual relationship detection (Relationship Det.) and predicate detection (Predicate Det.) measured by Top-100 recall (R@100) and Top-50 recall (R@50).

Method	Phrase Det.		Relationship Det.		Zero-Shot Phrase Det.		Zero-Shot Relationship Det.	
	R@100	R@50	R@100	R@50	R@100	R@50	R@100	R@50
CLC (CCA+Size+Position) [Plummer et al., 2016]	20.70	16.89	18.37	15.08	**15.23**	**10.86**	**13.43**	**9.67**
VTransE [Zhang et al., 2017a]	22.42	19.42	15.20	14.07	3.51	2.65	2.14	1.71
Vip-CNN [Li et al., 2017a]	**27.91**	22.78	20.01	17.32	-	-	-	-
VRL [Liang et al., 2017a]	22.60	21.37	20.79	18.19	10.31	9.17	8.52	7.94
Faster-RCNN + (AP+C+CAT)	25.26	23.88	23.39	20.14	11.28	10.73	10.17	9.57
Faster-RCNN + (AP+C+CAT) + Language Priors	25.56	**24.04**	**23.52**	**20.35**	11.30	10.78	10.26	9.54

Table 6.2: Results for visual relationship detection on the visual relationship benchmark dataset. Notice that we simply replace the detector with Faster-RCNN to extract a set of candidate object proposals without end-to-end jointly training the detector [Zhang et al., 2017a; Li et al., 2017a; Liang et al., 2017a] with the proposed method. And in CLC [Plummer et al., 2016], they use features and detection results from a Faster RCNN trained on external MSCOCO [Lin et al., 2014b] dataset and additional cues (e.g. size and position) are incorporated.

Comparison with [Sadeghi and Farhadi, 2011] and [Lu et al., 2016]: Finally, we compare our methods with the methods in [Sadeghi and Farhadi, 2011] and [Lu et al., 2016]. As seen, our methods achieve better performance than these two competing methods. Since our methods use the same object detection in [Lu et al., 2016], our result is most comparable to it. Note that our model does not employ explicit language priors modeling as in [Lu et al., 2016] and our improvement purely comes from the visual cue. This again demonstrates the power of context-aware interaction recognition.

To better evaluate our approach, we further visualize some test examples of AP+C+CAT in Figure 6.4. We can see that our predictions are reasonable in most cases.

6.4.1.2 Zero-shot learning performance evaluation

An important motivation of our method is to make the interaction classifier generalizable to unseen combinations of the interaction and context. In this section, we report the performance of our method on a zero-shot learning setting. Specifically, we train our models on the training set and evaluate their interaction classification performance on the 1877 unseen visual relationships in the test set. The results are reported in Table 7.5. From the table, we can see that the proposed methods work especially well in the zero-shot learning. For example, our best performed method (AP+C+CAT) almost doubled the performance on predicate detection in comparison with the Language Priors [Lu et al., 2016] method. This big improvement can be largely attributed to the advantage of using the context-aware scheme to model the interaction. In the Language Priors [Lu et al., 2016] method, the visual term for recognizing interaction is context-independent. Without context information to constrain the appearance variations, the learned interaction classifier tends to overfit the training set and fails to generalize to images with unseen interaction-context combinations. In comparison, with context-aware modeling, we explicitly consider the visual appearance variations introduced by changing context, thus more accurate and generalizable interaction classifier can be learned.

One interesting observation made in Table 7.5 is that the spatial feature representation produces better performance than the appearance based representation, as is evident from the superior performance of Spatial+C over AP methods. We speculate this is because spatial relationship features are more object independent and are less prone to overfiting the training set.

To intuitively evaluate zero-shot performance, we add some test examples of AP+C+CAT in Figure 6.5. We can make reasonable predictions on unseen interaction-context combinations in most cases.

Method	**Predicate Det.**		Phrase Det.		Relationship Det.	
	R@100	R@50	R@100	R@50	R@100	R@50
Language Priors [Lu et al., 2016]	8.45	8.45	3.75	3.36	3.52	3.13
Baseline1-app	7.44	7.44	3.08	2.82	2.91	2.74
Baseline1-spatial	7.27	7.27	2.14	2.14	2.14	2.14
Baseline2-app	7.36	7.36	2.22	1.71	2.05	1.54
Baseline2-spatial	0.43	0.43	0.09	0.09	0.09	0.09
Spatial+C	**16.42**	**16.42**	6.24	5.82	5.65	5.30
AP+C	15.06	15.06	5.82	5.05	5.22	4.62
AP+C+AT	15.00	15.00	5.62	5.02	5.36	4.76
AP+C+CAT	16.37	16.37	**6.59**	**5.99**	**5.99**	**5.47**

Table 6.3: Results for zero-shot visual relationship detection on the visual relationship benchmark dataset.

6.4.1.3 Extensions and comparison with state-of-the-art methods

Since the main focus of above experiments is to validate the advantage of the proposed methods over four competing baselines, we did not explore some techniques which could potentially further improve the visual relationship detection performance on the VRD dataset. To make our method achieve more comparable performance on the visual relationship and visual phrase detection tasks, we may consider two straightfoward extensions for our method: (1) use a better detector and (2) incorporate the language term trained in [Lu et al., 2016]. In the following part, we will examine the performance attained by applying these extensions and compare the resultant performance against the very latest state-of-the-art approaches [Liang et al., 2017a; Li et al., 2017a; Zhang et al., 2017a; Plummer et al., 2016] on the VRD dataset.

Improved detector: We first examine the effect of using a better detector by replacing the detection results obtained in [Lu et al., 2016] with that obtained by a Faster-RCNN detector [Girshick, 2015]. Note that the Faster-RCNN detector has also been used in [Liang et al., 2017a; Li et al., 2017a; Zhang et al., 2017a; Plummer et al., 2016] and using it will make our method comparable with the current state-of-the-arts. In our implementation, only the top 50 candidate object proposals, ranked by objectness confidence scores are extracted for mining relationships in per test image. The result of this modification is reported in Table 6.2 with our method annotated as Faster-RCNN + (AP+C+CAT). As seen, our method achieves best performance on phrase

detection R@50, relationship detection, zero-shot phrase and relationship detection. Note that our method can be further incorporated into the end-to-end relationship detection framework such as [Li et al., 2017a] to achieve even better performance.

Language priors: Language priors make significant contribution to [Lu et al., 2016] and in this section we apply the language priors released by [Lu et al., 2016] to investigate its impact. Following [Lu et al., 2016], we multiply our best performed model Faster-RCNN + (AP+C+CAT) with the language priors for interactions to obtain the final detection scores and the result is shown in Table 6.2 with the annotation Faster-RCNN + (AP+C+CAT) + Language Priors. Interestingly, the introduction of the language priors only introduces a marginal performance improvement. We suspect that is due to that our method builds a classifier with the information of both the interaction and context, and the correlation of interaction and context has been implicitly encoded. Therefore adding the language priors does not bring further benefit.

Figure 6.4: Qualitative examples of interaction recognition. We only predict the interaction between the ground-truth context bounding boxes. The phrases in the green bounding boxes are predicted while the phrases shown in the red bounding boxes are ground-truth.

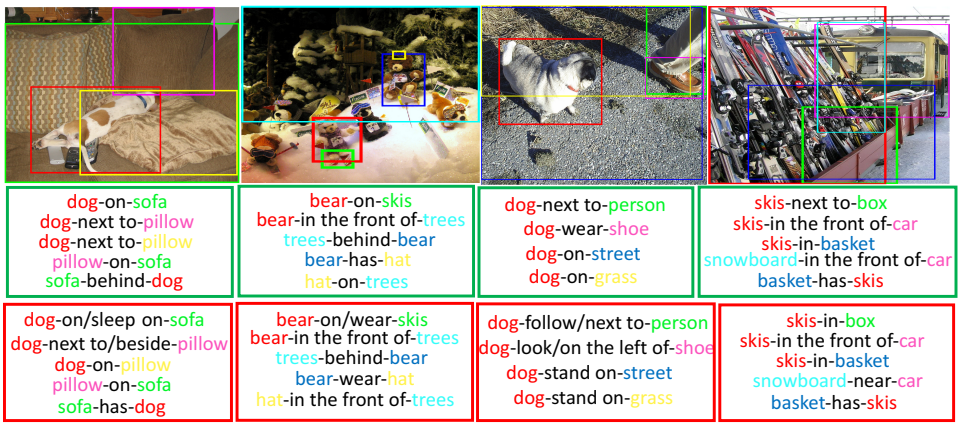


Figure 6.5: Qualitative examples of zero-shot interaction recognition. We only predict the interaction between the ground-truth context bounding boxes. The phrases in the green bounding boxes are predicted while the phrases shown in the red bounding boxes are ground-truth.

Method	Phrase Detection		Zero-Shot Phrase Detection	
	R@100	R@50	R@100	R@50
Visual Phrase [Sadeghi and Farhadi, 2011]	52.7	49.3	-	-
Language Priors [Lu et al., 2016]	82.7	78.1	23.9	11.4
Baseline1-app	70.1	65.6	12.4	10.5
Baseline1-spatial	68.3	63.6	10.3	8.9
Baseline2-app	77.5	72.3	11.0	9.2
Baseline2-spatial	15.7	10.4	1.1	0.5
Spatial+C	84.9	80.8	27.6	15.7
AP+C	85.9	81.6	28.5	16.4
AP+C+AT	86.2	82.1	28.8	17.9
AP+C+CAT	**86.8**	**82.9**	**30.2**	**18.7**

Table 6.4: Comparison of performance on the Visual Phrase dataset.

6.4.2 Evaluation on the visual phrase dataset

Following [Lu et al., 2016], we also run additional experiments on the Visual Phrase [Sadeghi and Farhadi, 2011] dataset. It has 17 phrases, out of which 12 of these phrases can be represented as triplet relationships as in the VRD dataset. We use the setting of [Lu et al., 2016] to conduct the experiment and report the R@50 and R@100 results in Table 6.4. Since the Visual Phrase dataset does not provide detection results, we apply the RCNN [Girshick et al., 2014] model to produce a set of candidate object regions and corresponding detection scores. As seen from Table 6.4, AP+C+CAT again achieves the best performance. In comparison with the performance of [Lu et al.,

2016], our method improves most in the zero-shot learning setting. This is consistent with the observation made in Sec. 6.4.1.2.

6.5 Summary

In this chapter, we study the role of context in recognizing the object interaction pattern. After identifying the importance of using context information, we propose a context-aware interaction classification framework which is accurate, scalable and enjoys good generalization ability to recognize unseen context-interaction combinations. Further, we investigate various ways to derive the visual represention for interaction patterns and extend the context-aware framework to design a new attention-pooling layer. With extensive experiments, we validate the advantage of the proposed methods and produce the state-of-the-art performance on two visual relationship detection datasets.

Chapter 7

HCVRD: a benchmark for large-scale Human-Centered Visual Relationship Detection

Statement of Authorship

Title of Paper	HCVRD: a benchmark for large-scale Human-Centered Visual Relationship Detection
Publication Status	☐ Published ☒ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted w ork w ritten in manuscript style
Publication Details	Accepted for publication in AAAI conference on Artificial Intelligence (AAAI), 2018.

Principal Author

Name of Principal Author (Candidate)	Bohan Zhuang
Contribution to the Paper	Wrote the paper and completed the experiments.
Overall percentage (%)	90%
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.
Signature	Date 7/12/2017

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Qi Wu
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/2017

Name of Co-Author	Chunhua Shen
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/2017

Please cut and paste additional co-author panels here as required.

Name of Co-Author	Ian Reid
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 7/12/17

Name of Co-Author	Anton van den Hengel
Contribution to the Paper	Helped design the method and modified the paper.
Signature	Date 18/12/17

7.1 Overview

Visual relationship detection aims to capture interactions between pairs of objects in images. Relationships between objects and humans represent a particularly important subset of this problem, with implications for challenges such as understanding human behavior, and identifying affordances, amongst others. In addressing this problem we first construct a large-scale human-centric visual relationship detection dataset (HCVRD), which provides many more types of relationship annotations (nearly 10K categories) than the previous released datasets. This large label space better reflects the reality of human-object interactions, but gives rise to a long-tail distribution problem, which in turn demands a zero-shot approach to labels appearing only in the test set. This is the first time this issue has been addressed. We propose a webly-supervised approach to these problems and demonstrate that the proposed model provides a strong baseline on our HCVRD dataset.

7.2 Introduction

The challenge in visual relationship detection [Li et al., 2017a; Liang et al., 2017a; Lu et al., 2016] is to capture interactions between pairs of objects in an image. In this chapter, rather than detect interactions between arbitrary objects, we focus on capturing the relationships between a human and an object. Recognising human-object relationships is a problem of significant practical import, and a subtly different challenge, than the more general case. Humans have a far wider variety of modes of interaction than general objects, but they also have agency, meaning that more can be drawn from human-object interactions than from other interactions. For example, a human can interact with a bicycle in multiple ways (such as carry, hold, ride, park, push *etc.*), but the relationships between bicycles and other objects are far simpler. The human interactions also imply intent, and possibly provide information about the past or future that is typically lacking from object-object relationships. Previous

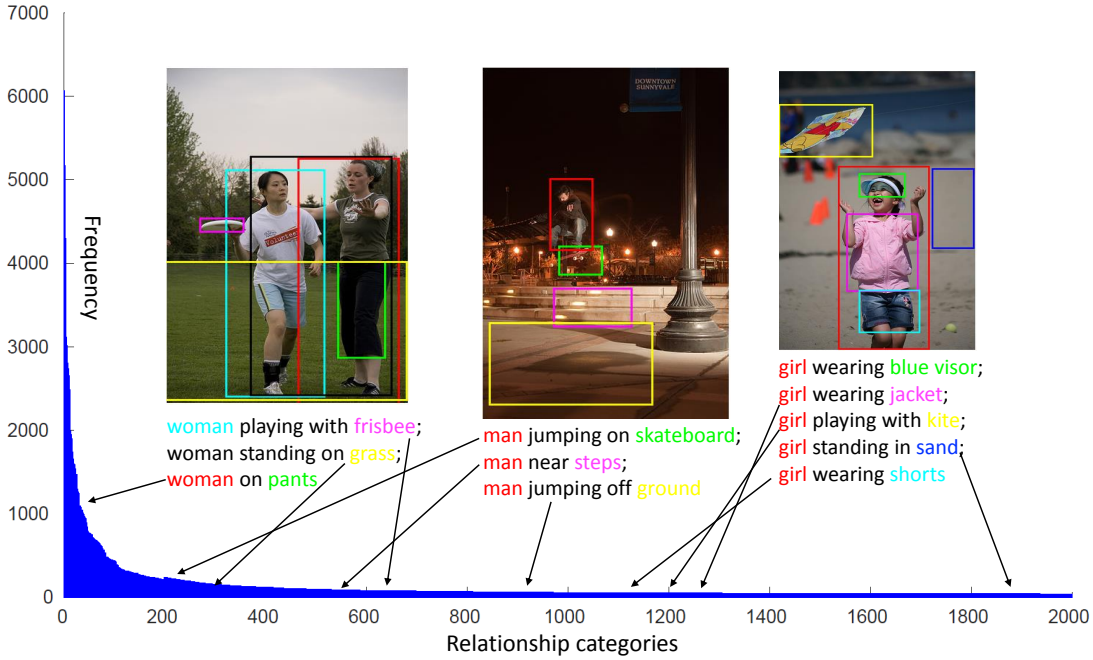


Figure 7.1: The long-tail label distribution of our HCVRD dataset. We only show the top-2000 relationships because the tail is too long. Three example images are also shown, with our webly-supervised model detected results. The color of human and objects in the phrases correspond to the color of the bounding boxes. The arrows indicate the 'location' of the relationship in the label distribution. As we can see, most of the relationships are lie on the tail. Some of them such as 'girl wearing blue visor' is not even in the top-2000.

work [Chao et al., 2017, 2015] has similarly recognised that human-object interactions of particular interest, and have proposed several datasets.

As in so many problems of practical interest, the label space of realistic human-centric visual relationship detection (HCVRD) exhibits a long tail distribution, meaning that there are very few, to zero, training examples for the vast majority of labels. This is a fundamental problem for the standard deep learning approach, which relies on large numbers of examples for each class. If deep learning is to progress from easy, and often artificially simplified problems for which copious training data is available, datasets will need to better reflect the practical reality of the majority of problems. The main contribution of this chapter is a large-scale human-centric visual relationships detection (HCVRD) dataset , which accurately depicts the long-tail label distribution of the problem, thus necessitating zero-shot recognition.

Datasets	#relationships (no zero-shot)	#predicates	#objects	#images	#zero-shot relationships
Verbs-COCO [Gupta and Malik, 2015]	-	26	80	10346	-
Stanford 40 actions [Yao et al., 2011]	40	35	28	9532	-
MPII Human Pose [Andriluka et al., 2014]	410	-	66	40522	-
HICO-DET [Chao et al., 2017]	520	117	80	47774	-
Ours	9852	927	1824	52855	18471

Table 7.1: Comparison of the existing human-object interaction detection datasets.

We formulate the human-centric visual relationships detection problem as that of detecting relationship triplets $\langle human, predicate, object \rangle$ in the image, with bounding boxes on the human subject and object. The HCVRD dataset is constructed based on the Visual Genome [Krishna et al., 2017]. Compared to the previous *human-object interaction* works [Chao et al., 2017, 2015], there are several differences. First, we have more fine-grained labels. For the 'human' item in the triplet, we are not satisfied only detecting a 'human' subject, instead, we have four sub-categories which are man(adult), woman(adult), boy and girl. This is valuable because the gender and age can affect the way that a human interact with objects. For example, we are unlikely to find 'a man holding a Barbie' but this relationship is more commonly seen for 'a girl'. Except for the 'human' type, our 'predicate' covers a much wider range of 'relationships' than the 'interactions' in the previous setting. The dataset contains 9852 different relationships, nearly 20 times more than the HICO dataset [Chao et al., 2015]. Such a big label space leads to a long-tail label distribution, i.e., some labels appear less than 10 times. Additionally, we provide 18,471 zero-shot relationships, i.e., relationships that never appear in the training split. To the best of our knowledge, this is the biggest dataset with these two forms of labels provided and that is labeled with both human-centric visual relationships and corresponding 'human' and 'object' bounding boxes.

Motivated by above challenges, our second contribution is developing methods for (i) automatically augmenting the training set using weakly labeled data crawled from the web; and (ii) performing zero-shot recognition by comparing the query data to web-retrieved data. While not radically novel in approach, our methods address the issues raised in long-tail datasets and provide, we believe, a strong baseline for further works based on our HCVRD dataset and similar data.

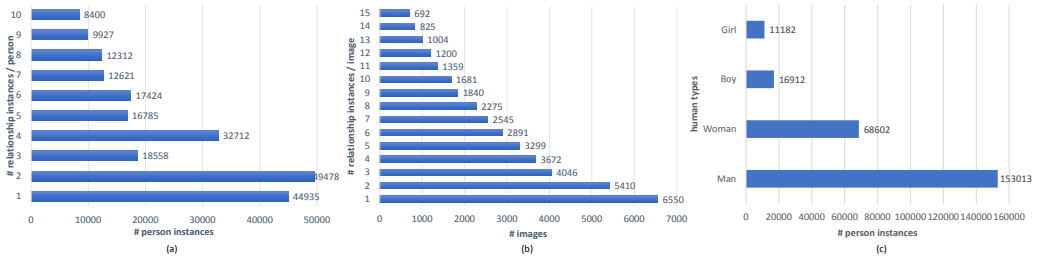


Figure 7.2: Statistics of the HCVRD dataset, the distribution of the (a): number of different relationships that occur on a person. (b): number of relationships in each image. (c): human types.

7.3 The HCVRD Dataset

Our dataset comprises two parts, publicly available separately or together from `Hiddenforblindreview`. The main part comprises a carefully curated set harvested from the large Visual Genome dataset [Krishna et al., 2017]. In addition we have created a supplementary component of 788,160 images drawn from the top 100 image-search results for each relationship triple.

7.3.1 Constructing HCVRD dataset

Our proposed human-centric visual relationship detection (HCVRD) dataset is constructed based on the Visual Genome dataset [Krishna et al., 2017], which provides detailed scene annotations, such as objects, attributes and relationships (defined as {sub, predicate, obj}). Since we are only interested in the relationships involving human subjects, the first step is to extract all the human-related relationships from the 2.3 million relationships pool in the Visual Genome [Krishna et al., 2017]. This is done automatically by searching all the relationships that their 'subject' include a 'human' concept (we use the WordNet [Leacock and Chodorow, 1998] to define a 'human' concept vocabulary including human, person, people, man, male, woman, boy, girl *etc.*.)

It is worth noting that there are some relationships that only appear once in the dataset. We annotate a 'zero-shot' tag on those labels so that they can test under the zero-shot setting. This is one of the significant differences with previous human-

object interaction dataset, such as the HICO [Chao et al., 2017]. The zero-shot setting can verify the generalization ability of an algorithm, i.e., the ability to detect unseen relationships in the training set.

The collected relationships are still noisy and should be carefully processed. We first manually correct the annotations that contain misspellings and noisy characters (e.g. comma). We then eliminate the attribute predicates (such as "has", "is", "are") because these predicates are too abstract and may lead to a weak discriminative model. We further normalize the predicates by eliminating the tense using a lexical analysis toolkit [Bird et al., 2009] and finally have 927 predicate categories, which cover a wide range of types, such as action, spatial, preposition, comparative and verb and so on. We then merge some semantically similar objects by using the GloVe [Pennington et al., 2014] (i.e., two words are merged if their similarity calculated based on the Global Vector words representation is bigger than a threshold) and normalize (singularization and eliminate the article) the remaining object names while keeping their fine-grained attributes (e.g. black shirt, yellow shirt). Furthermore, we manually divide the 'human' subject into four more fine-grained classes according to the image content, which are man(adult), woman(adult), boy and girl. This is a valuable setting because the gender and age can affect the way that the human interacts with objects.

7.3.2 Dataset Statistics

Table 7.1 provides summary statistics about our proposed HCVRD dataset, compared with some human-object interactions dataset. In the following part, we highlight several interesting aspects of the data.

We finally have 52,855 images with 1,824 object categories and 927 predicates. In total, the dataset contains 256,550 relationships instances with 9,852 non zero-shot relationship types and 18,471 zero-shot relationships types. There are on average 10.63 predicates per object category. We use 31,586 images for training and construct

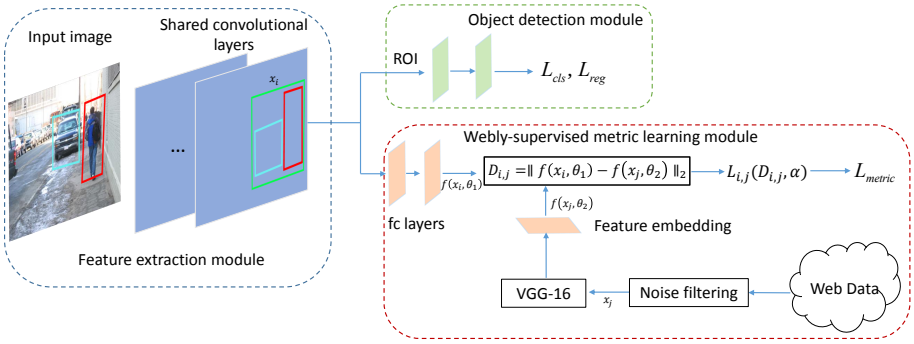


Figure 7.3: The framework of the proposed model. The model consists of (a):a feature extraction module, (b): an object detection module, (c) a webly-supervised metric learning module. The three modules can be jointly trained in an end-to-end manner.

two test splits. The first test split contains 10,000 images where all the relationships occur in the training set. Another test split includes all the zero-shot relationships, i.e., relationships in this split are never occurred in the training split. The distribution of human-object relationships in our dataset (see Figure 7.1) highlight the long-tail effect of infrequent relationships. Specifically, there are 370 relationships that appear more than 100 times and 7,474 relationships appear fewer than 10 times.

Figure 7.2 (a) shows a distribution of the number of different relationship instances that occurred on a person. Unlike past datasets where each person only can have one relationship, each people in the HCVRD dataset has on average 2.62 relationships with other objects. Figure 7.2 (b) shows the distribution of number of relationship instances in each image. Our HCVRD dataset has a large number of images with more than one relationship instance. On average there are 6.13 relationship instances annotated per image. Figure 7.2 (c) shows the distribution of human types (such as man, woman, boy and girl) in our dataset.

7.3.3 Supplementary web data

In addition to the curated main dataset described above we have collected a supplementary set of 788,160 images which are also available for download, and which we use in our model for metric learning, to provide a baseline for recognition in long-tailed data. To collect these images we automatically crawl using Google Im-

ages as the source of candidates. We treat all the 9,852 relationships as the query list and process each category independently, taking the top 100 images returned as representing that relationship class.

For most basic categories commonly appearing in the visual world, the top results returned by Google image search are quite clean so that we can directly learn useful visual representations from them. However some returned images may have wildly different content from the query triple, and this can adversely affect training of the model. To mitigate this issue, we employ the weakly-supervised noise robust approach of [Zhuang et al., 2017a] to filter the noisy images fully automatically.

More specifically, [Zhuang et al., 2017a] relies on a random group training process that randomly groups multiple web data (images) into a single training instance as the input of a classification neural network (we use a separate network for this purpose, performing 1-of-9,852 classification). As the size of the group increases, the chances diminish exponentially that a training instance (i.e. a group) does not contain imagery of the true relationship. [Zhuang et al., 2017a] shows that this simple "trick" can lead to sizeable gains in accuracy when training with weakly labelled data. To determine which image or images from a group contain true positive imagery, an attentional pooling layer is employed on the last convolutional layer to determine which neuron activations have contributed to the classification. More specifically, we use the attention weights to decide a confidence score for each individual image in the random group. We then sort all images of a given relationship category according to their confidence scores, and retain the top 80% (discarding the remaining 20%). This process yields a relatively clean (though still weakly labelled) set of supplementary data that covers the entire set of 9,852 relationship categories with 80 images per category (hence 788,160).

7.4 A webly-supervised model

One of the biggest challenges in our proposed dataset is the long-tail distribution of the labels. Nearly 80% of the relationship labels in our dataset have fewer than 10 training examples. This issue creates a big challenge to the conventional supervised learning models, especially for those deep convolutional neural network based models, which normally require a large number of examples to train. Part of our purpose in creation of the dataset is to stimulate research in this important direction. To this end, we propose a strong baseline model for recognition in long-tailed data based on a so-called "webly"-supervised learning approach. Such an approach aims to leverage (practically) unlimited weakly labelled web data to overcome the restriction of limited training examples and the long-tail distribution.

An overview of our proposed webly-supervised relationship detection (WSRD) model is shown in Figure 7.3. Our model is divided into three sub-modules: the feature extraction module, the detection module and the distance metric learning module. The feature extraction module is a stack of convolutional layers and max-pooling layers which have the same configuration as the VGG-16 [Simonyan and Zisserman, 2015] or the ResNet [He et al., 2016a]. The detection module is in the style of Faster-RCNN [Ren et al., 2015], which is used to detect the object and human subject (in its sub-category). A bounding box that encompasses the detected human-object pair (i.e, contains both human and object) is sent to a deep metric learning module, which performs inference by finding the nearest-neighbour match in the web-crawled data amongst all triples sharing the same human and object labels. This determines the predicate category. The neighbourhood distances are computed using the learned distance metric (i.e. in the feature space).

The three sub-modules can be learned in an end-to-end manner. For the efficiency, the feature map generated by the feature extraction module is shared as input to following two modules. We use the VGG-16 [Simonyan and Zisserman, 2015] network as a basic building block for our model. We discuss the detection module and the

Method	Predicate Det.				Phrase Det.				Relationship Det.			
	R@50		R@100		R@50		R@100		R@50		R@100	
	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3
Multilabel	0.87	2.78	0.87	2.78	0.44	0.92	0.50	0.95	0.03	0.07	0.04	0.09
JointCNN	2.68	7.36	2.68	7.36	2.35	5.63	2.39	6.14	0.21	0.44	0.22	0.53
SeparateCNN	29.00	44.37	29.00	45.87	8.24	10.53	8.92	13.81	0.48	0.60	0.50	0.66
Ours	31.08	47.66	31.08	48.98	10.03	13.05	10.75	16.94	0.53	0.68	0.59	0.72

Table 7.2: Evaluation of different methods on the proposed dataset. The results reported include visual relationship detection (Relationship Det.) and predicate detection (Predicate Det.) measured by Top-100 recall (R@100) and Top-50 recall (R@50).

Method	Predicate Det.				Phrase Det.				Relationship Det.			
	R@50		R@100		R@50		R@100		R@50		R@100	
	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3
Multilabel	0.45	1.09	0.45	1.09	0.22	0.58	0.24	0.62	0.01	0.01	0.01	0.01
JointCNN	0.02	0.03	0.02	0.03	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
SeparateCNN	15.94	26.73	15.94	26.73	0.49	1.55	0.58	1.96	0.04	0.08	0.05	0.10
Ours-without web data	18.01	29.35	18.01	29.35	0.73	2.15	0.80	2.43	0.06	0.10	0.07	0.13
Ours	24.55	36.59	24.55	36.59	1.76	3.62	1.91	4.56	0.12	0.16	0.14	0.21

Table 7.3: Results for human-object relationship detection on the long-tail benchmark subset.

distance metric learning module in more detail in the following sections.

7.4.1 Detection module

The object (and human subject) detection module structure is identical to that of the Faster-RCNN [Ren et al., 2015]. Taking the output of the feature extraction module (`Conv5_3` feature map) as the input, the Region Proposal Network (RPN) is used to generate object proposals. During training, we extract features with RoIPool for each object proposal, followed by the bounding box regression loss L_{reg} and a classification loss L_{cls} to learn the detector/classifier in a manner identical to [Ren et al., 2015]. During inference, we use this module to detect all human subjects and objects in the images. We apply non-maximum suppression (NMS) to reduce the number of proposals with the IoU (Intersection of Union) threshold 0.3 and objectiveness scores higher than 0.2. These filtered boxes are further grouped to all possible $\langle human, object \rangle$ pairs and a bounding box that fully contains the human and object boxes is associated to each pair. These "union" bounding boxes are (separately and individually) the input to the distance metric learning module.

Method	Predicate Det.				Phrase Det.				Relationship Det.			
	R@50		R@100		R@50		R@100		R@50		R@100	
	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3
(a) Without metric learning module	22.55	33.12	22.55	33.87	5.87	7.33	6.04	9.44	0.29	0.43	0.34	0.49
(b) Without noise filtering	30.36	46.12	30.37	46.68	9.92	12.96	10.67	16.36	0.49	0.64	0.57	0.70
(c) Ours (full model)	31.08	47.66	31.08	48.98	10.03	13.05	10.75	16.94	0.53	0.68	0.59	0.72

Table 7.4: Ablation studies on the HCVRD benchmark non-zeroshot test set.

Method	Predicate Det.				Phrase Det.				Relationship Det.			
	R@50		R@100		R@50		R@100		R@50		R@100	
	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3	top-1	top-3
Multilabel	-	-	-	-	-	-	-	-	-	-	-	-
JointCNN	-	-	-	-	-	-	-	-	-	-	-	-
SeparateCNN	2.75	4.98	2.99	5.93	0.06	0.11	0.07	0.16	0.01	0.05	0.03	0.08
Ours	8.15	12.34	8.57	13.42	0.88	1.43	0.92	1.84	0.03	0.09	0.05	0.12

Table 7.5: Results for human-object relationship detection on the zero-shot benchmark test set.

7.4.2 Distance metric learning module

As noted above, this module accepts a union region of the detected human and object, and computes the feature-space distance between the proposed region and all of the web-crawled visual relationship data. The nearest class label of the web data is assigned to the proposed region. The distance metric function is learned via deep metric learning on the web-crawled (supplementary) data.

More specifically, the deep metric learning process aims to learn a semantic feature embedding (a feature space) for which similar examples are mapped close to one another while dissimilar examples are mapped further apart. To this end, we construct a set of positive pairs and a set of negative pairs by drawing from the main dataset and the web data. Each positive pair $(\mathbf{x}_i, \mathbf{x}_j)$ contains a sample from the main HCVRD dataset and a sample from the web data with the same label, while each negative pair is similarly drawn one from each, but with non-matching labels. We follow [Oh Song et al., 2016] to incrementally add the positive and negative pairs. Specifically, we first sample a few anchor pairs and then active mining hard negative images to the batch, more details can be found in [Oh Song et al., 2016].

During the training, the ground truth predicate region $\mathbf{x}_i$'s corresponding `Conv5_3` feature map is used as part of the input for the metric learning module. In the inference, we first detect the human and objects and get all the possible union bounding boxes' corresponding `Conv5_3` feature map as the input, separately and individually.

Then the convolutional feature map is sent to two fully connected layers and the output $f(x_i, \theta_1)$ serves as part of the input for the metric learning functions (see equation (7.1)), where f is the feed-forward function and θ_1 is the learnable parameters of the feature extraction module with the fully connected layers. Another input $f(x_j, \theta_2)$ of the metric learning functions is from the collected web data, which is passed through a pre-trained VGG-16 model and a learnable feature embedding layer with parameter θ_2. Following [Oh Song et al., 2016], the metric is then learned using a structured loss function based on the sampled positive and negative pairs of training samples:

$$\begin{aligned} L_{mec} &= \frac{1}{2|\mathbb{P}|} \sum_{(i,j)\in\mathbb{P}} \max(0,\ L_{i,j})^2, \\ L_{i,j} &= \log\Big(\sum_{(i,k)\in\mathbb{N}} \exp(\alpha - D_{i,k}) + \sum_{(j,l)\in\mathbb{N}} \exp(\alpha - D_{j,l})\Big) + D_{i,j} \end{aligned} \tag{7.1}$$

where $\mathbb{P}$ is the set of positive pairs and $\mathbb{N}$ is the set of negative pairs, $D_{i,j} = \left\| f(x_i, \theta_1) - f(x_j, \theta_2) \right\|_2$ is distance between two embedding feature vectors. The α is the learnable margin parameter.

The two modules can be jointly trained in an end-to-end manner. The model employs multi-task loss for human-object relationship detection:

$$L = L_{reg} + L_{cls} + L_{mec} \tag{7.2}$$

where L_{reg} and L_{cls} are the regression loss and cross-entropy loss in the detection module.

7.5 Experiments

7.5.1 Implementation details

We set the feature embedding size in the metric learning module as 256. For training efficiency, we initialize the feature extraction module with the pre-trained VGG-16. We then pretrained the detection module and fix it while training the metric learning module. The learning rate is initialized to 0.0001 and decreased by a factor of 10 after

every 5 epochs. During the inference, we first retrieve the top 20 nearest neighbor relationships and select those including both detected human and object categories. Then we use the top-ranked selected candidates for evaluation.

7.5.2 Evaluation Setup

We evaluate our human-object interactions task using Recall@100 and Recall@50, following the setting of Visual Relationship Detection (VRD) task [Liang et al., 2017a; Lu et al., 2016]. Recall@x computes the fraction of times the correct relationship is calculated in the top x predictions, which are ranked by the final distances. We evaluate on three tasks: (1) For **predicate detection**, the goal is to predict the accuracy of *predicate* recognition, where the groundtruth labels and bounding boxes for both the *object* and *human* are given. (2) In **phrase detection**, we aim to predict ⟨*human-predicate-object*⟩ and localize the entire relationship in one bounding box. (3) For **relationship detection**, the task is to recognize ⟨*human-predicate-object*⟩ and localize both human and object bounding boxes, where both boxes should have at least 0.5 overlap (IoU) with the ground-truth in order to be regarded as correct prediction. In the real world applications, different relationships may share very similar semantic meanings (e.g. "man holding phone", "man talking on phone", "man using phone") and it's difficult to differentiate them. Hence, in many cases, one "appropriate" prediction may be judged "incorrect" due to the limitation of the test annotations, which is a common problem of the current VRD evaluation metric. One possible solution is to employ the human evaluation, which is cost however. In this chapter, we instead report both top-1 and top-3 results under different Recalls to evaluate the model.

7.5.3 Baselines

We benchmark the following approaches on our new dataset and results are reported in Table 7.2.

Multilabel classification A person can concurrently perform different interac-

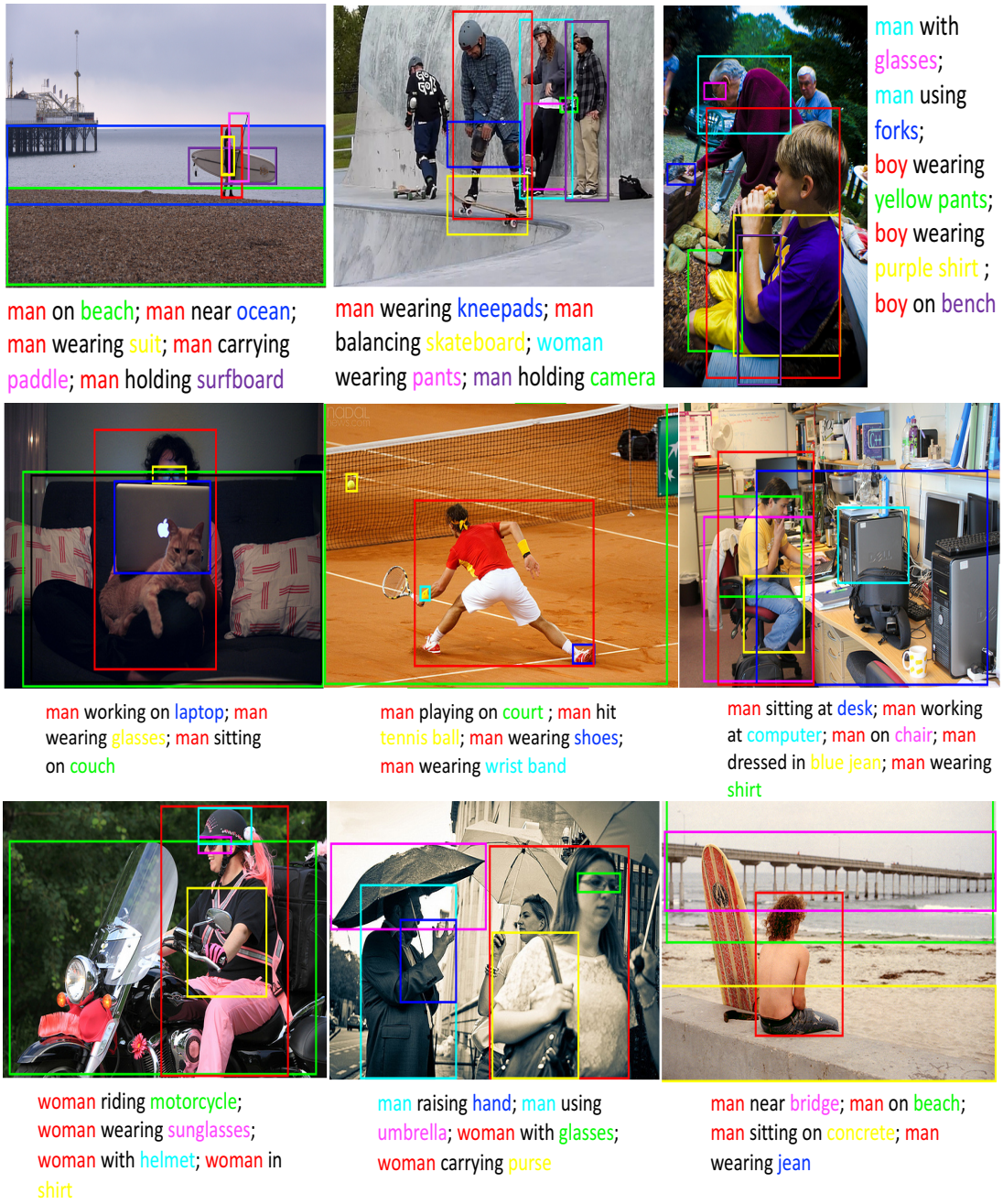


Figure 7.4: Qualitative examples of the predicate detection. The color of human and objects in the phrases correspond to the color of the bounding boxes. We only predict the interactions between the ground-truth bounding box pairs.

tions with different target objects, e.g. a person can "ride bicycle" and "drink water" at the same time. Thus we treat the human-object relationship detection task as a multilabel classification problem where we apply a sigmoid cross entropy loss on

Figure 7.4: Qualitative examples of the predicate detection. The color of human and objects in the phrases correspond to the color of the bounding boxes. We only predict the interactions between the ground-truth bounding box pairs.

top of the classification layer. Specifically, we treat the union of a human and its correlated objects as the input during training. During the testing, we use our object detection module to return the regions. We use VGG-16 model as the basis building block.

JointCNN This implements the Visual phrases [Sadeghi and Farhadi, 2011]. We train a VGG-16 model to jointly predict the three components of a relationship. Specifically, we treat each relationship category separately and train a 9,852 way classification model.

SeparateCNN Following the visual model of [Lu et al., 2016], we first train a VGG-

16 model to classify the 1,824 objects. Similarly, we train a second model to classify each of the 927 predicates using the union of the bounding boxes of the participating human and the object in that relationship.

For *JointCNN* and *Multilabel* baselines, we empirically find that due to the long-tail property of the dataset, the learned models are seriously biased. It causes the predictions only fall into those labels with large numbers of training examples. To solve the problem of extreme classification with enormous number of categories, we instead propose to employ the metric learning approach with web data to perform efficient nearest neighbor inference on the learned metric space. By comparing *ours* with the two baselines, we find significant performance increase on all evaluation metrics.

For the *SeperateCNN* baseline, since the training data for human, objects and predicates are relatively adequate respectively, its performance is competitive with our proposed method. In other words, the human, objects and predicates are predicted separately, hence, the label prediction space is much smaller than above two baseline approaches. However, compared to predicate detection results, the performance of phrase and relationship detection decreases a lot. It shows that detecting such wide range of objects is a major challenge for visual relationship detection. We also show some qualitative examples in Figure 7.4.

7.5.4 Long-tail evaluation

Due to the long-tail distribution of the categories in the dataset, the infrequent relationships will contribute not much to the final testing performance. But in real world applications, the relationships in long-tail should not be ignored. So we select those relationships that appear less than 10 times as a subset (i.e. there are totally 7,474 relationships) and report the performance in Table 7.3. From the table, we can see that our approach performs steadily better than the baseline methods. For the baseline methods, the lack of training data is a main challenge for obtaining accurate predic-

tions. The main motivation of the proposed method is to utilize web data to tackle this limitation. With the always available web data, we can learn the distance metrics and efficiently infer nearest neighbor relationships on the learned metric space.

7.5.5 Ablation study

With vs. without metric learning module Metric learning module is the key component of our system. To evaluate its impact, we implement a variant without the metric learning module. For the detected union bounding boxes of relationships and web data, we directly extract the 4096-dimentional feature vector for each sample using the pretrained VGG-16 model. We then compute the cosine similarity between the test sample and all mean vectors of the relationship categories that contain both detected human and object types. We then retrieve the nearest neighbor relationship categories as our predictions. Table 7.4 (a) vs. (c) shows that learning the semantic feature embeddings via distance metric contributes a lot to the final performance.

With vs. without web data We also evaluate the influence of the web data by only using the training data of the dataset. Since one motivation of introducing web data is to solve the scarceness of training data, we report this variant under the long-tail setting in Table 7.3 as *Ours-without web data*. By comparing it with *Ours* in Table 7.3, we find that removing web data causes an obvious performance degradation, which proves the effectiveness of introducing the web data. We find that the web data can help on some relationships that rarely happened in the dataset, such as 'man cooking on street' and 'man peddling rickshaw'.

With vs. without noise filtering We further remove the noise filtering step to investigate the affect of noisy labels. The results are shown in Table 7.4 (b). Table 7.4 (b) vs. (c) shows that removing noise filtering have less affect to the performance compared to removing metric learning module. This is because for relationships that commonly used in the visual content, top results returned by Google images search are pretty clean. Noise filtering provides an auxiliary to further improve the quality

of web data.

7.5.6 Zero-shot evaluation

It is quite important to make the model generalizable to unseen human-object relationships. In this section, we report the performance of our method on a zero-shot learning setting. Specifically, we train our models on the training set and evaluate their relationship detection performance on the 18,471 unseen visual relationships in the zero-shot test split. Given the detected human and objects in a relationship, we first get all their possible interactions to form a search space. We then collect web data and extract feature embeddings to get the nearest neighbors relationships for the test sample. The results are reported in Table 7.5. We can see that the proposed method works more robust. This can be attributed to the introduction of the external web data for efficient nearest neighbor search. For the "separateCNN" baseline, by predicting the predicates separately from its objects, it is difficult to capture the appearance variations due to the weak and even ambiguous visual features.

7.6 Summary

We have proposed a large-scale human-centric visual relationship detection (HCVRD) dataset, which is significantly larger and broader than previous datasets. Human-centric relationships represent an important subclass of all relationships, not only because the human has agency, but also due to their practical importance for other challenges. Increasing the scale of data available better captures the reality of the task, but rises two important practical problems, the long-tail distribution issue and the zero-shot problem, which are both reflected in our proposed HCVRD dataset. Motivated by the practical importance of the task, our webly-supervised method addresses the issues and provides a strong baseline for further works based on our HCVRD dataset and similar data.

Chapter 8

Conclusion and future work

8.1 Conclusion

In this thesis, we study how to build energy-efficient deep learning algorithms for real applications. What's more, we also focus on dealing with two important issues existing in visual recognition. Specifically, we addressed the following problems:

- As the network grows deeper, the model complexity will increase exponentially in both the training and testing stages, which leads to very high demand in computing resources. To solve this problem, in Chapter 3, we have proposed three effective approaches to solve the optimization problem for low-bitwidth deep neural networks. The first approach is a two-step optimization strategy, that is to quantize the weights and activations separately. We also observed that continuously quantizing from high-precision to low-precision is also beneficial to the final performance. To better utilize the knowledge from the full-precision model, we have also proposed joint learning of the low-precision model and its full-precision counterpart. The three approaches can be used jointly or separately. We show that even using only 4-bit weights and activations for all layers, we can outperform the 32-bit model on ImageNet and Cifar100 with either AlexNet or ResNet-50.

- To realize efficient data storage and fast image search, in Chapter 4, we have proposed a novel hashing method to learn a mapping from the image space to compact binary space. Specifically, to solve the extremely high computational

complexity in the triplet space, we have proposed to formulate high-order binary codes learning as a multi-label classification problem by explicitly separating learning into two interleaved stages. We have improved the training speed by two-orders of magnitude and the hashing performance on several retrieval datasets.

- The success of deep learning in visual recognition applications largely relies on massive datasets, which are quite hard and expensive to obtain. To solve this problem, in Chapter 5, we have proposed an end-to-end weakly-supervised deep learning framework which is robust to the label noise in Web images. Specifically, we have proposed to apply the attention mechanism in a random grouping framework which can effectively filter the noise from both in-correctly labeled images and less discriminative image regions. We have also collected a fine-grained car dataset from web images and the superior performance of the proposed method is demonstrated.

- Another issue of visual recognition is the higher-level understanding of the scene. As an intermediate level task connecting the image caption and object detection, we focus on solving the visual relationship/phrase detection task in Chapter 6 and Chapter 7. Specifically, in Chapter 6, we have explicitly constructed a context-aware classifier which combines the context, and the interaction. As a result, such an interaction classifier can be adapted to its context which can improve the zero-shot generalization abilities. In addition, our simple framework is robust to various feature representations and show improved performance. And in Chapter 7, since recognizing human-object relationships is an important component of visual relationship detection, we have further proposed a large-scale human-centric visual relationship detection dataset (HCVRD), which provides many more types of relationship annotations (nearly 10K categories) than the previously released datasets. To solve the long-tail distribution problem in the label space, based on the noise-robust method

in Chapter 5, we have proposed a webly-supervised approach and demonstrate that the proposed model provides a strong baseline on our HCVRD dataset.

8.2 Future Work

In addition to the problems addressed in this thesis, we also point out the following open problems that we expect to explore in the future:

- In Chapter 3, to further improve the quantization performance at 2-bit, we can replace the general quantization function to the ones that more suitable to the ternary weights and activations (i.e., [Zhu et al., 2017] use two full precision scaling coefficients in each layer). What's more, we can combine the quantization method with low-rank approximation, architecture design and other network compression methods.

- For the triplet-based hashing method proposed in Chapter 4, we can improve the triplets sampling strategy for selecting more discriminative hard negative samples and positive samples. What's more, inspired by the progressive quantization method proposed in Chapter 3, we can employ the progressive quantization technique on the binary layer to further improve the performance. Furthermore, we can propose a complete pipeline that consists of quantized neural network and binary codes generation strategy for faster nearest neighbor search.

- For the web learning approach proposed in Chapter 5, we can further improve our approach by adding an extra noise layer into the network which adapts outputs of the network to match the distribution of noisy label. What's more, we can add one more batch normalization layer after the attention pooling layer to solve the possible scale problem in the proposed framework.

- To further investigate the visual relationship detection task in Chapter 6 and Chapter 7, we can expand the triplet relationship (i.e., subject-predicate-object)

to structured scene graphs where nodes denote detected objects and edges depict their relationships. Then we can use message passing or other graph optimization techniques to solve the dense graph problem.

Bibliography

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B., 2014. 2d human pose estimation: New benchmark and state of the art analysis. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3686–3693. (cited on page 114)

Arnold, A.; Nallapati, R.; and Cohen, W. W., 2007. A comparative study of methods for transductive transfer learning. In *International Conference on Data Mining Workshops*, 77–82. IEEE. (cited on page 94)

Ba, J. and Caruana, R., 2014. Do deep nets really need to be deep? In *Proc. Adv. Neural Inf. Process. Syst.*, 2654–2662. (cited on pages 19, 20, and 26)

Bastien, F.; Lamblin, P.; Pascanu, R.; Bergstra, J.; Goodfellow, I.; Bergeron, A.; Bouchard, N.; Warde-Farley, D.; and Bengio, Y., 2012. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, (2012). (cited on pages 57 and 77)

Beigman, E. and Klebanov, B. B., 2009. Learning with annotation noise. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL*, 280–287. ACL. (cited on page 12)

Bengio, Y.; Léonard, N.; and Courville, A., 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, (2013). (cited on page 23)

Berg, T.; Liu, J.; Woo Lee, S.; Alexander, M. L.; Jacobs, D. W.; and Belhumeur, P. N., 2014. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2011–2018. (cited on page 3)

Bilen, H.; Fernando, B.; Gavves, E.; Vedaldi, A.; and Gould, S., 2016. Dynamic image networks for action recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3034–3042. (cited on page 91)

Bird, S.; Klein, E.; and Loper, E., 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.". (cited on page 116)

Brodley, C. E. and Friedl, M. A., 2011. Identifying mislabeled training data. *arXiv preprint arXiv:1106.0219*, (2011). (cited on page 12)

Carreira-Perpinan, M. A. and Raziperchikolaei, R., 2015. Hashing with binary autoencoders. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 557–566. (cited on page 10)

Chao, Y.-W.; Liu, Y.; Liu, X.; Zeng, H.; and Deng, J., 2017. Learning to Detect Human-Object Interactions. *arXiv preprint arXiv:1702.05448*, (2017). (cited on pages 113, 114, and 116)

Chao, Y.-W.; Wang, Z.; He, Y.; Wang, J.; and Deng, J., 2015. Hico: A benchmark for recognizing human-object interactions in images. In *Proc. IEEE Int. Conf. Comp. Vis.*, 1017–1025. (cited on pages 113 and 114)

Chen, D.; Cao, X.; Wang, L.; Wen, F.; and Sun, J., 2012. Bayesian face revisited: A joint formulation. In *Proc. Eur. Conf. Comp. Vis.*, 566–579. (cited on page 62)

Chen, X. and Gupta, A., 2015. Webly supervised learning of convolutional networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1431–1439. (cited on page 11)

Chen, X.; Shrivastava, A.; and Gupta, A., 2013. Neil: Extracting visual knowledge from web data. In *Proc. IEEE Int. Conf. Comp. Vis.*, 1409–1416. (cited on page 11)

Choi, W.; Chao, Y.-W.; Pantofaru, C.; and Savarese, S., 2013. Understanding indoor scenes using 3d geometric phrases. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 33–40. (cited on page 13)

Chollet, F., 2016. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint arXiv:1610.02357*, (2016). (cited on pages 9 and 22)

Chua, T.-S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; and Zheng, Y., 2009. Nus-wide: a real-world web image database from national university of singapore. In *Proc. of the ACM Int. Conf. on Image and Video Retrieval.* (cited on page 56)

Courbariaux, M.; Bengio, Y.; and David, J.-P., 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Proc. Adv. Neural Inf. Process. Syst.*, 3123–3131. (cited on pages 8, 19, and 21)

Denton, E. L.; Zaremba, W.; Bruna, J.; LeCun, Y.; and Fergus, R., 2014. Exploiting linear structure within convolutional networks for efficient evaluation. In *Proc. Adv. Neural Inf. Process. Syst.*, 1269–1277. (cited on pages 8 and 21)

Desai, C. and Ramanan, D., 2012. Detecting actions, poses, and objects with relational phraselets. *Proc. Eur. Conf. Comp. Vis.*, (2012), 158–172. (cited on page 13)

Desai, C.; Ramanan, D.; and Fowlkes, C. C., 2011. Discriminative models for multi-class object layout. *Int. J. Comp. Vis.*, 95, 1 (2011), 1–12. (cited on page 91)

Divvala, S. K.; Farhadi, A.; and Guestrin, C., 2014. Learning everything about anything: Webly-supervised visual concept learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3270–3277. (cited on page 11)

Do, C. and Ng, A. Y., 2005. Transfer learning for text classification. In *Proc. Adv. Neural Inf. Process. Syst.*, 299–306. (cited on page 94)

Erin Liong, V.; Lu, J.; Wang, G.; Moulin, P.; and Zhou, J., 2015. Deep hashing for compact binary codes learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2475–2483. (cited on page 10)

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A., 2010. The pascal visual object classes (voc) challenge. *Int. J. Comp. Vis.*, 88, 2 (2010), 303–338. (cited on page 68)

Evgeniou, T. and Pontil, M., 2004. Regularized multi–task learning. In *International Conference on Data Mining*, 109–117. ACM. (cited on page 94)

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32, 9 (2010), 1627–1645. (cited on page 12)

Fergus, R.; Fei-Fei, L.; Perona, P.; and Zisserman, A., 2010. Learning object categories from internet image searches. *Proceedings of the IEEE*, 98, 8 (2010), 1453–1466. (cited on page 11)

Gionis, A.; Indyk, P.; Motwani, R.; et al., 1999. Similarity search in high dimensions via hashing. In *Proc. Int. Conf. Very Large Datadases*, vol. 99, 518–529. (cited on page 9)

Girshick, R., 2015. Fast r-cnn. In *Proc. IEEE Int. Conf. Comp. Vis.*, 1440–1448. (cited on pages 12 and 105)

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 580–587. (cited on pages 7, 12, 68, 79, and 107)

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F., 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35, 12 (2013), 2916–2929. (cited on pages xiii, 10, 56, and 58)

Guo, Y.; Yao, A.; and Chen, Y., 2016. Dynamic network surgery for efficient dnns. In *Proc. Adv. Neural Inf. Process. Syst.*, 1379–1387. (cited on pages 9 and 22)

Gupta, A. and Davis, L. S., 2008. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proc. Eur. Conf. Comp. Vis.*, 16–29. Springer. (cited on page 91)

Gupta, S. and Malik, J., 2015. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, (2015). (cited on page 114)

Han, S.; Mao, H.; and Dally, W. J., 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Int. Conf. Learn. Rpresen.*, (2016). (cited on pages 9, 19, and 22)

Han, S.; Pool, J.; Tran, J.; and Dally, W., 2015. Learning both weights and connections for efficient neural network. In *Proc. Adv. Neural Inf. Process. Syst.*, 1135–1143. (cited on pages 9, 18, and 22)

He, K.; Gkioxari, G.; Dollar, P.; and Girshick, R., 2017. Mask r-cnn. In *Proc. IEEE Int. Conf. Comp. Vis.* (cited on page 3)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2014. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Proc. Eur. Conf. Comp. Vis.*, 346–361. Springer. (cited on page 11)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. IEEE Int. Conf. Comp. Vis.*, 1026–1034. (cited on page 11)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016a. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 770–778. (cited on pages 7, 9, 11, 18, 20, 21, 68, and 119)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016b. Identity mappings in deep residual networks. In *Proc. Eur. Conf. Comp. Vis.*, 630–645. Springer. (cited on pages 9, 20, and 21)

Hinton, G.; Vinyals, O.; and Dean, J., 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, (2015). (cited on pages 19, 20, and 26)

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, (2012). (cited on page 11)

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, (2017). (cited on pages 9 and 22)

Hu, R.; Rohrbach, M.; Andreas, J.; Darrell, T.; and Saenko, K., 2017. Modeling relationships in referential expressions with compositional modular networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on pages 91 and 96)

Huang, G. B.; Ramesh, M.; Berg, T.; and Learned-Miller, E., 2007. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst. (cited on page 62)

Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; and Bengio, Y., 2016. Binarized neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 4107–4115. (cited on pages 20 and 23)

Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; and Keutzer, K., 2016. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, (2016). (cited on pages 9 and 21)

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. Int. Conf. Mach. Learn.*, 448–456. (cited on page 29)

Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; and Darrell, T., 2014. Caffe: Convolutional architecture for fast feature

embedding. In *Proc. of the ACM Int. Conf. on Multimedia.*, 675–678. (cited on page 62)

Jiang, K.; Que, Q.; and Kulis, B., 2015. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 4933–4941. (cited on page 9)

Karpathy, A. and Fei-Fei, L., 2015. Deep visual-semantic alignments for generating image descriptions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3128–3137. (cited on page 91)

Kim, Y.-D.; Park, E.; Yoo, S.; Choi, T.; Yang, L.; and Shin, D., 2015. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530*, (2015). (cited on pages 8, 19, and 21)

Kingma, D. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, (2014). (cited on page 99)

Klare, B. F.; Klein, B.; Taborsky, E.; Blanton, A.; Cheney, J.; Allen, K.; Grother, P.; Mah, A.; Burge, M.; and Jain, A. K., 2015. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1931–1939. (cited on pages xvii, 61, and 63)

Krause, J.; Sapp, B.; Howard, A.; Zhou, H.; Toshev, A.; Duerig, T.; Philbin, J.; and Fei-Fei, L., 2016. The unreasonable effectiveness of noisy data for fine-grained recognition. In *Proc. Eur. Conf. Comp. Vis.*, 301–320. Springer. (cited on pages 11 and 79)

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; et al., 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comp. Vis.*, 123, 1 (2017), 32–73. (cited on pages 114 and 115)

Krizhevsky, A., 2009. Learning multiple layers of features from tiny images. Technical report. (cited on page 56)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 1097–1105. (cited on pages 3, 18, 30, and 68)

Kulis, B. and Darrell, T., 2009. Learning to hash with binary reconstructive embeddings. In *Proc. Adv. Neural Inf. Process. Syst.*, 1042–1050. (cited on page 10)

Kulis, B. and Grauman, K., 2009. Kernelized locality-sensitive hashing for scalable image search. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2130–2137. (cited on page 9)

Kumar, M. P. and Koller, D., 2010. Efficiently selecting regions for scene understanding. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3217–3224. (cited on page 13)

Lai, H.; Pan, Y.; Liu, Y.; and Yan, S., 2015. Simultaneous feature learning and hash coding with deep neural networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3270–3278. (cited on pages xiii, xvii, 10, 46, 54, 56, 57, 58, and 60)

Leacock, C. and Chodorow, M., 1998. Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database*, (1998). (cited on page 115)

Lebedev, V. and Lempitsky, V., 2016. Fast convnets using group-wise brain damage. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2554–2564. (cited on pages 9 and 22)

Li, X.; Lin, G.; Shen, C.; Van den Hengel, A.; and Dick, A., 2013. Learning hash functions using column generation. In *Proc. Int. Conf. Mach. Learn.*, 142–150. (cited on pages 10 and 46)

Li, Y.; Ouyang, W.; Wang, X.; et al., 2017a. Vip-cnn: Visual phrase guided convolutional neural network. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 7244–7253. (cited on pages xvii, 13, 91, 92, 95, 103, 105, 106, and 112)

Li, Y.; Ouyang, W.; Zhou, B.; Wang, K.; and Wang, X., 2017b. Scene graph generation from objects, phrases and region captions. In *Proc. IEEE Int. Conf. Comp. Vis.* (cited on page 92)

Liang, X.; Lee, L.; and P. Xing, E., 2017a. Deep Variation-structured Reinforcement Learning for Visual Relationship and Attribute Detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on pages xvii, 92, 95, 103, 105, 112, and 123)

Liang, X.; Lee, L.; and Xing, E. P., 2017b. Deep variation-structured reinforcement learning for visual relationship and attribute detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on page 13)

Lin, G.; Shen, C.; Shi, Q.; van den Hengel, A.; and Suter, D., 2014a. Fast supervised hashing with decision trees for high-dimensional data. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1971–1978. (cited on pages xiii, 5, 46, 47, 49, 50, 52, 53, 56, and 58)

Lin, G.; Shen, C.; Suter, D.; and van den Hengel, A., 2013. A general two-step approach to learning-based hashing. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2552–2559. (cited on pages 46, 47, and 48)

Lin, G.; Shen, C.; van den Hengel, A.; and Reid, I., 2016a. Efficient piecewise training of deep structured models for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3194–3203. (cited on page 3)

Lin, G.; Shen, C.; van den Hengel, A.; and Reid, I., 2016b. Efficient piecewise training of deep structured models for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on page 68)

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L., 2014b. Microsoft coco: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, 740–755. Springer. (cited on pages xvii, 68, and 103)

Liu, B.; Wang, M.; Foroosh, H.; Tappen, M.; and Pensky, M., 2015. Sparse convolutional neural networks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 806–814. (cited on pages 9 and 22)

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C., 2016. Ssd: Single shot multibox detector. In *Proc. Eur. Conf. Comp. Vis.*, 21–37. Springer. (cited on page 68)

Liu, W.; Wang, J.; Ji, R.; Jiang, Y.-G.; and Chang, S.-F., 2012. Supervised hashing with kernels. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2074–2081. (cited on pages xiii, 10, 56, and 58)

Liu, W.; Wang, J.; Kumar, S.; and Chang, S.-F., 2011. Hashing with graphs. In *Proc. Int. Conf. Mach. Learn.*, 1–8. (cited on pages 10 and 57)

Long, J.; Shelhamer, E.; and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440. (cited on pages 7 and 68)

Lu, C.; Krishna, R.; Bernstein, M.; and Fei-Fei, L., 2016. Visual relationship detection with language priors. In *Proc. Eur. Conf. Comp. Vis.*, 852–869. Springer. (cited on pages 13, 91, 92, 101, 102, 103, 104, 105, 106, 107, 112, 123, and 125)

Lu, J.; Xiong, C.; Parikh, D.; and Socher, R., 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on page 91)

Ma, L.; Lu, Z.; Shang, L.; and Li, H., 2015. Multimodal convolutional neural networks for matching image and sentence. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2623–2631. (cited on page 91)

Maas, A. L.; Hannun, A. Y.; and Ng, A. Y., 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. Int. Conf. Mach. Learn.*, vol. 30. (cited on page 11)

Mallya, A. and Lazebnik, S., 2016. Learning models for actions and person-object interactions with transfer to question answering. In *Proc. Eur. Conf. Comp. Vis.*, 414–428. Springer. (cited on page 91)

Manwani, N. and Sastry, P., 2013. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43, 3 (2013), 1146–1151. (cited on page 12)

Mathias, M.; Benenson, R.; Pedersoli, M.; and Van Gool, L., 2014. Face detection without bells and whistles. In *Proc. Eur. Conf. Comp. Vis.*, 720–735. (cited on page 62)

Miranda, A. L.; Garcia, L. P. F.; Carvalho, A. C.; and Lorena, A. C., 2009. Use of classification algorithms in noise detection and elimination. In *International Conference on Hybrid Artificial Intelligence Systems*, 417–424. Springer. (cited on page 12)

Mnih, V. and Hinton, G. E., 2012. Learning to label aerial images from noisy data. In *Proc. Int. Conf. Mach. Learn.*, 567–574. (cited on page 11)

Niu, L.; Li, W.; and Xu, D., 2015. Visual recognition by learning from web data: A weakly supervised domain generalization approach. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2774–2783. (cited on page 11)

Norouzi, M.; Blei, D. M.; and Salakhutdinov, R. R., 2012. Hamming distance metric learning. In *Proc. Adv. Neural Inf. Process. Syst.*, 1061–1069. (cited on page 46)

Novikov, A.; Podoprikhin, D.; Osokin, A.; and Vetrov, D. P., 2015. Tensorizing neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 442–450. (cited on pages 8 and 21)

Oh Song, H.; Xiang, Y.; Jegelka, S.; and Savarese, S., 2016. Deep metric learning via lifted structured feature embedding. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 4004–4012. (cited on pages 121 and 122)

Parameswaran, S. and Weinberger, K. Q., 2010. Large margin multi-task metric learning. In *Proc. Adv. Neural Inf. Process. Syst.*, 1867–1875. (cited on page 94)

Parisotto, E.; Ba, J. L.; and Salakhutdinov, R., 2016. Actor-mimic: Deep multitask and transfer reinforcement learning. *Int. Conf. Learn. Rpresen.*, (2016). (cited on pages 19 and 26)

Patricia, N. and Caputo, B., 2014. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1442–1449. (cited on page 94)

Pennington, J.; Socher, R.; and Manning, C., 2014. Glove: Global vectors for word representation. 1532–1543. (cited on page 116)

Plummer, B. A.; Mallya, A.; Cervantes, C. M.; Hockenmaier, J.; and Lazebnik, S., 2016. Phrase localization and visual relationship detection with comprehensive linguistic cues. *arXiv preprint arXiv:1611.06641*, (2016). (cited on pages xvii, 13, 95, 96, 103, and 105)

Quattoni, A. and Torralba, A., 2009. Recognizing indoor scenes. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 413–420. (cited on page 56)

Ramanathan, V.; Li, C.; Deng, J.; Han, W.; Li, Z.; Gu, K.; Song, Y.; Bengio, S.; Rosenberg, C.; and Fei-Fei, L., 2015. Learning semantic relationships for better action retrieval in images. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1100–1109. (cited on page 91)

Rastegari, M.; Ordonez, V.; Redmon, J.; and Farhadi, A., 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *Proc. Eur. Conf. Comp. Vis.*, 525–542. (cited on page 20)

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 779–788. (cited on pages 3 and 13)

Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A., 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, (2014). (cited on page 11)

Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 91–99. (cited on pages 7, 12, 68, 119, and 120)

Rohrbach, A.; Rohrbach, M.; Hu, R.; Darrell, T.; and Schiele, B., 2016. Grounding of textual phrases in images by reconstruction. In *Proc. Eur. Conf. Comp. Vis.*, 817–834. Springer. (cited on page 91)

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y., 2015. Fitnets: Hints for thin deep nets. *Int. Conf. Learn. Rpresen.*, (2015). (cited on pages 19, 20, and 26)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L., 2015a. Imagenet large scale visual recognition challenge. *Int. J. Comp. Vis.*, (2015), 1–42. (cited on pages 56, 68, and 77)

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al., 2015b. Imagenet large scale visual recognition challenge. *Int. J. Comp. Vis.*, 115, 3 (2015), 211–252. (cited on pages 31 and 99)

Russell, B. C.; Freeman, W. T.; Efros, A. A.; Sivic, J.; and Zisserman, A., 2006. Using multiple segmentations to discover objects and their extent in image collec-

tions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, vol. 2, 1605–1614. (cited on page 13)

Sadeghi, F.; Kumar Divvala, S. K.; and Farhadi, A., 2015. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1456–1464. (cited on page 91)

Sadeghi, M. A. and Farhadi, A., 2011. Recognition using visual phrases. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1745–1752. (cited on pages 13, 102, 103, 107, and 125)

Schroff, F.; Criminisi, A.; and Zisserman, A., 2011. Harvesting image databases from the web. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33, 4 (2011), 754–766. (cited on page 11)

Schroff, F.; Kalenichenko, D.; and Philbin, J., 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 815–823. (cited on pages 46, 60, and 63)

Shen, F.; Shen, C.; Liu, W.; and Shen, H. T., 2015. Supervised discrete hashing. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 37–45. (cited on page 10)

Shen, F.; Shen, C.; Shi, Q.; Van Den Hengel, A.; and Tang, Z., 2013. Inductive hashing on manifolds. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1562–1569. (cited on page 10)

Simonyan, K. and Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Rpresen.* (cited on pages 11, 18, 55, 62, 68, 97, and 119)

Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R., 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, (2014). (cited on page 12)

Sukhbaatar, S. and Fergus, R., 2015. Learning from noisy labels with deep neural networks. In *Int. Conf. Learn. Rpresen. Workshops*. (cited on pages 11 and 83)

Sun, Y.; Liang, D.; Wang, X.; and Tang, X., 2015. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, (2015). (cited on page 63)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going deeper with convolutions. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1–9. (cited on pages 9, 11, and 21)

Tai, C.; Xiao, T.; Zhang, Y.; Wang, X.; et al., 2015. Convolutional neural networks with low-rank regularization. *arXiv preprint arXiv:1511.06067*, (2015). (cited on pages 8 and 21)

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1701–1708. (cited on page 63)

Van Horn, G.; Branson, S.; Farrell, R.; Haber, S.; Barry, J.; Ipeirotis, P.; Perona, P.; and Belongie, S., 2015. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 595–604. (cited on page 3)

Wang, D.; Otto, C.; and Jain, A. K., 2015a. Face search at scale: 80 million gallery. *arXiv preprint arXiv:1507.07242*, (2015). (cited on pages 61 and 63)

Wang, J.; Song, Y.; Leung, T.; Rosenberg, C.; Wang, J.; Philbin, J.; Chen, B.; and Wu, Y., 2014. Learning fine-grained image similarity with deep ranking. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1386–1393. (cited on pages 46 and 57)

Wang, L.; Qiao, Y.; and Tang, X., 2015b. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 4305–4314. (cited on page 91)

Weinberger, K. Q. and Saul, L. K., 2009. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10 (2009), 207–244. (cited on page 60)

Weiss, Y.; Fergus, R.; and Torralba, A., 2012. Multidimensional spectral hashing. In *Proc. Eur. Conf. Comp. Vis.*, 340–353. (cited on page 10)

Weiss, Y.; Torralba, A.; and Fergus, R., 2009. Spectral hashing. In *Proc. Adv. Neural Inf. Process. Syst.*, 1753–1760. (cited on page 10)

Wen, W.; Wu, C.; Wang, Y.; Chen, Y.; and Li, H., 2016. Learning structured sparsity in deep neural networks. In *Proc. Adv. Neural Inf. Process. Syst.*, 2074–2082. (cited on pages 9 and 22)

Wu, J.; Leng, C.; Wang, Y.; Hu, Q.; and Cheng, J., 2016a. Quantized convolutional neural networks for mobile devices. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 4820–4828. (cited on page 20)

Wu, Q.; Shen, C.; Liu, L.; Dick, A.; and van den Hengel, A., 2016b. What value do explicit high level concepts have in vision to language problems? In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 203–212. (cited on page 91)

Wu, Q.; Wang, P.; Shen, C.; Dick, A.; and van den Hengel, A., 2016c. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 4622–4630. (cited on page 91)

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X., 2015. Learning from massive noisy labeled data for image classification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2691–2699. (cited on pages 11, 12, 79, and 83)

Xiong, X. and De la Torre, F., 2013. Supervised descent method and its applications to face alignment. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 532–539. (cited on page 62)

Xu, D.; Zhu, Y.; Choy, C. B.; and Fei-Fei, L., 2017. Scene graph generation by iterative message passing. *arXiv preprint arXiv:1701.02426*, (2017). (cited on page 13)

Xu, Z.; Huang, S.; Zhang, Y.; and Tao, D., 2015. Augmenting strong supervision using web data for fine-grained categorization. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2524–2532. (cited on page 11)

Yang, L.; Luo, P.; Change Loy, C.; and Tang, X., 2015. A large-scale car dataset for fine-grained categorization and verification. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 3973–3981. (cited on page 76)

Yao, B.; Jiang, X.; Khosla, A.; Lin, A. L.; Guibas, L.; and Fei-Fei, L., 2011. Human action recognition by learning bases of action attributes and parts. In *Proc. IEEE Int. Conf. Comp. Vis.*, 1331–1338. (cited on page 114)

Yi, D.; Lei, Z.; Liao, S.; and Li, S. Z., 2014. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, (2014). (cited on pages 62 and 63)

Zagoruyko, S. and Komodakis, N., 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*, (2016). (cited on pages 9, 11, and 21)

Zagoruyko, S. and Komodakis, N., 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *Int. Conf. Learn. Rpresen.*, (2017). (cited on pages 19, 26, and 29)

Zeiler, M. D. and Fergus, R., 2014. Visualizing and understanding convolutional networks. In *Proc. Eur. Conf. Comp. Vis.*, 818–833. (cited on page 11)

Zhang, B.; Wang, L.; Wang, Z.; Qiao, Y.; and Wang, H., 2016a. Real-time action recognition with enhanced motion vector cnns. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on page 91)

Zhang, H.; Kyaw, Z.; Chang, S.-F.; and Chua, T.-S., 2017a. Visual translation embedding network for visual relation detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on pages xvii, 95, 96, 103, and 105)

Zhang, H.; Kyaw, Z.; Chang, S.-F.; and Chua, T.-S., 2017b. Visual translation embedding network for visual relation detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on page 13)

Zhang, R.; Lin, L.; Zhang, R.; Zuo, W.; and Zhang, L., 2015. Bit-scalable deep hashing with regularized similarity learning for image retrieval. *IEEE Trans. Image Proc.*, , 12 (2015), 4766–4779. (cited on page 46)

Zhang, X.; Zhou, X.; Lin, M.; and Sun, J., 2017c. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *arXiv preprint arXiv:1707.01083*, (2017). (cited on pages 9 and 22)

Zhang, X.; Zou, J.; He, K.; and Sun, J., 2016b. Accelerating very deep convolutional networks for classification and detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38, 10 (2016), 1943–1955. (cited on pages 8, 19, and 21)

Zhang, Y.; Xiang, T.; Hospedales, T. M.; and Lu, H., 2017d. Deep mutual learning. *arXiv preprint arXiv:1706.00384*, (2017). (cited on pages 19, 26, and 38)

Zhao, F.; Huang, Y.; Wang, L.; and Tan, T., 2015. Deep semantic ranking based hashing for multi-label image retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 1556–1564. (cited on pages 10, 46, and 48)

Zhao, R.-W.; Li, J.; Chen, Y.; Liu, J.-M.; Jiang, Y.-G.; and Xue, X., 2016. Regional gating neural networks for multi-label image classification. In *Proc. Brit. Mach. Vis. Conf.* (cited on page 78)

Zhou, A.; Yao, A.; Guo, Y.; Xu, L.; and Chen, Y., 2017. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv preprint arXiv:1702.03044*, (2017). (cited on pages 8, 19, 21, and 31)

Zhou, S.; Wu, Y.; Ni, Z.; Zhou, X.; Wen, H.; and Zou, Y., 2016. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, (2016). (cited on pages 19, 20, 23, 30, and 31)

Zhu, C.; Han, S.; Mao, H.; and Dally, W. J., 2017. Trained ternary quantization. *Int. Conf. Learn. Rpresen.*, (2017). (cited on pages 8, 19, 21, 30, 31, 33, and 131)

Zhuang, B.; Liu, L.; Li, Y.; Shen, C.; and Reid, I., 2017a. Attend in Groups: A Weakly-Supervised Deep Learning Framework for Learning From Web Data. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (cited on page 118)

Zhuang, B.; Liu, L.; Shen, C.; and Reid, I., 2017b. Towards context-aware interaction recognition for visual relationship detection. In *Proc. IEEE Int. Conf. Comp. Vis.* (cited on page 13)