Document Viewer

Turnitin Originality Report

Processed on: 12-Aug-2019 15:40 HKT
ID: 1158043036
Word Count: 28421
Submitted: 4

Image Understanding from Imperfect Data via T... By Ge Weifeng

Similarity Index	Similarity by Source	
78%	Internet Sources:	75%
	Publications:	74%
	Student Papers:	22%

exclude quoted exclude bibliography exclude small matches download
print mode: quickview (classic) report ▼ Change mode

25% match (Internet from 14-May-2019)
https://www.groundai.com/project/multi-evidence-filtering-and-fusion-for-multi-label-classification-object-detection-and-semantic-segmentation-based-on-weakly-supervised-learning/

17% match (Internet from 07-Sep-2017)
http://i.cs.hku.hk

13% match (Internet from 11-Apr-2019)
https://arxiv.org/pdf/1810.06951.pdf

6% match (publications)
"Computer Vision – ECCV 2018", Springer Nature America, Inc, 2018

3% match (publications)
Weifeng Ge, Sibei Yang, Yizhou Yu. "Multi-evidence Filtering and Fusion for Multi-label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018

2% match (student papers from 05-Mar-2017)
Submitted to University of Hong Kong on 2017-03-05

1% match (Internet from 21-Sep-2018)
http://openaccess.thecvf.com

1% match (Internet from 12-Sep-2017)
http://www.scitepress.org

1% match (Internet from 21-Nov-2017)

http://hub.hku.hk

1% match (publications)
Weifeng Ge, Weilin Huang, Dengke Dong, Matthew R. Scott. "Chapter 17 Deep Metric Learning with Hierarchical Triplet Loss", Springer Nature America, Inc, 2018

1% match (Internet from 21-Nov-2017)
http://hub.hku.hk

1% match (Internet from 21-Sep-2018)
http://openaccess.thecvf.com

<1% match (Internet from 21-Sep-2018)
http://openaccess.thecvf.com

<1% match (publications)
Weifeng Ge, Yizhou Yu. "Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Fine-Tuning", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017

<1% match (student papers from 14-Jun-2017)
Submitted to University of Hong Kong on 2017-06-14

<1% match (Internet from 08-May-2019)
https://www.groundai.com/project/a-survey-of-recent-advances-in-texture-representation/1

<1% match (student papers from 19-Jan-2017)
Submitted to University of Edinburgh on 2017-01-19

<1% match (student papers from 05-Jan-2018)
Submitted to City University on 2018-01-05

<1% match (publications)
"Computer Vision – ACCV 2018", Springer Science and Business Media LLC, 2019

<1% match (Internet from 29-Aug-2017)
http://users.cecs.anu.edu.au

<1% match (student papers from 05-Sep-2017)
Submitted to University College London on 2017-09-05

<1% match (student papers from 08-Aug-2019)
Submitted to Sabanci Universitesi on 2019-08-08

<1% match (Internet from 20-Nov-2015)
http://biostats.bepress.com

<1% match (Internet from 04-Sep-2018)
http://openaccess.thecvf.com

<1% match (student papers from 12-Jan-2019)
Submitted to Heriot-Watt University on 2019-01-12

<1% match (student papers from 30-Nov-2017)
Submitted to University of Hong Kong on 2017-11-30

<1% match (student papers from 09-May-2016)
Submitted to Imperial College of Science, Technology and Medicine on 2016-05-09

<1% match (student papers from 13-Jan-2019)
Submitted to University of Edinburgh on 2019-01-13

<1% match (student papers from 22-Jul-2018)
Submitted to University of Hong Kong on 2018-07-22

<1% match (student papers from 27-Jun-2018)
Submitted to Indian Institute of Technology on 2018-06-27

<1% match (student papers from 12-Jul-2016)
Submitted to University of Alabama on 2016-07-12

<1% match (student papers from 01-Oct-2018)
Submitted to City University on 2018-10-01

<1% match (Internet from 17-May-2011)
http://cs-people.bu.edu

<1% match (student papers from 23-Apr-2018)
Submitted to University of Hong Kong on 2018-04-23

<1% match (student papers from 23-May-2017)
Submitted to University of Hong Kong on 2017-05-23

<1% match (student papers from 18-Oct-2018)
Submitted to University of Bristol on 2018-10-18

<1% match (student papers from 23-May-2019)
Submitted to Higher Education Commission Pakistan on 2019-05-23

<1% match (Internet from 26-Feb-2014)
http://www.clef-initiative.eu

<1% match (student papers from 28-Feb-2018)
Submitted to University of Illinois at Urbana-Champaign on 2018-02-28

<1% match (student papers from 15-May-2019)
Submitted to University of the Philippines Los Banos on 2019-05-15

<1% match (Internet from 08-Mar-2016)
http://arizona.openrepository.com

<1% match (Internet from 21-Sep-2018)
http://openaccess.thecvf.com

<1% match (student papers from 24-Nov-2017)
Submitted to University of Bath on 2017-11-24

<1% match (student papers from 11-Sep-2017)
Submitted to University of Kent at Canterbury on 2017-09-11

<1% match (Internet from 19-Jan-2016)
http://aaai.org

<1% match (student papers from 06-Jan-2019)
Submitted to University of Oxford on 2019-01-06

<1% match (student papers from 04-Mar-2018)
Submitted to National College of Ireland on 2018-03-04

<1% match (student papers from 19-Apr-2017)
Submitted to University of Sheffield on 2017-04-19

<1% match (student papers from 21-Jan-2016)
Submitted to University of Edinburgh on 2016-01-21

<1% match (Internet from 27-Jan-2014)
http://matthewkusner.com

<1% match (student papers from 05-Aug-2019)
Submitted to Özyegin Üniversitesi on 2019-08-05

<1% match (student papers from 23-Mar-2016)
Submitted to University of Hong Kong on 2016-03-23

<1% match (Internet from 21-Sep-2018)
http://openaccess.thecvf.com

<1% match (Internet from 14-Jul-2017)
http://ml.cs.tsinghua.edu.cn

<1% match (Internet from 27-Oct-2017)
http://www.asci.tudelft.nl

<1% match (student papers from 09-May-2016)
Submitted to Higher Education Commission Pakistan on 2016-05-09

<1% match (student papers from 28-May-2018)
Submitted to Loughborough University on 2018-05-28

<1% match (student papers from 25-Jan-2018)
Submitted to University of Edinburgh on 2018-01-25

<1% match (Internet from 08-Aug-2019)
http://export.arxiv.org

<1% match (publications)
"Computer Vision – ECCV 2016", Springer Science and Business Media LLC, 2016

<1% match (Internet from 11-Apr-2010)
http://etds.ncl.edu.tw

<1% match (student papers from 03-Sep-2015)
Submitted to University of Liverpool on 2015-09-03

<1% match (Internet from 04-Sep-2018)
https://arxiv.org/abs/1702.08690

<1% match (Internet from 01-Apr-2012)

http://en.wikipedia.org

<1% match (publications)
"Computer Vision – ECCV 2018", Springer Science and Business Media LLC, 2018

<1% match (publications)
Feifei Zhang, Tianzhu Zhang, Qirong Mao, Lingyu Duan, Changsheng Xu. "Facial Expression Recognition in the Wild", 2018 ACM Multimedia Conference on Multimedia Conference - MM '18, 2018

<1% match (student papers from 01-Feb-2019)
Submitted to Heriot-Watt University on 2019-02-01

<1% match (student papers from 07-Sep-2017)
Submitted to University College London on 2017-09-07

<1% match (student papers from 29-Jan-2015)
Submitted to University of Hong Kong on 2015-01-29

<1% match (Internet from 08-Jan-2019)
https://alexgkendall.com/media/papers/alex_kendall_phd_thesis_compressed.pdf

<1% match (student papers from 20-May-2019)
Submitted to National Research University Higher School of Economics on 2019-05-20

<1% match (student papers from 17-Mar-2018)
Submitted to Universitaet Dortmund Hochschulrechenzentrul on 2018-03-17

<1% match (publications)
Lecture Notes in Computer Science, 2015.

<1% match (Internet from 22-Oct-2010)
http://www.kecl.ntt.co.jp

<1% match (Internet from 08-Sep-2017)
https://tel.archives-ouvertes.fr/tel-01538307/file/TH_T2557_ytang.pdf

<1% match (student papers from 29-Aug-2017)
Submitted to University of Oxford on 2017-08-29

<1% match (publications)
"Computer Vision – ECCV 2018", Springer Nature America, Inc, 2018

<1% match (student papers from 30-Oct-2017)
Submitted to Higher Education Commission Pakistan on 2017-10-30

<1% match (student papers from 26-Aug-2013)
Submitted to King Fahd University for Petroleum and Minerals on 2013-08-26

<1% match (student papers from 05-Sep-2016)
Submitted to University College London on 2016-09-05

<1% match (Internet from 08-Feb-2014)
http://www.lxduan.info

<1% match (Internet from 31-May-2018)
http://docplayer.net

<1% match (Internet from 09-Feb-2019)
https://tel.archives-ouvertes.fr/tel-01797231/document

<1% match (Internet from 21-Mar-2010)
http://galciv.net

<1% match (student papers from 05-Apr-2018)
Submitted to University of Illinois at Urbana-Champaign on 2018-04-05

<1% match (student papers from 15-May-2018)
Submitted to National Technical University of Athens on 2018-05-15

<1% match (publications)
"Neural Information Processing", Springer Nature, 2017

<1% match (publications)
"Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications", Springer Science and Business Media LLC, 2019

<1% match (publications)
"Computer Vision – ECCV 2018", Springer Nature America, Inc, 2018

<1% match (student papers from 26-Nov-2018)
Submitted to University of Auckland on 2018-11-26

Abstract of thesis entitled "Image Understanding from Imperfect Data via Transfer Learning, Metric Learning, and Weakly Supervised Learning" Submitted by Weifeng GE for the degree of Doctor of Philosophy at The University of Hong Kong in Aug 2019 Huge amount of labeled data has led to a set of breakthroughs in image understanding, such as object/ scene recognition, object detection, semantic segmentation, and etc. To generate reliable deep models, rich data annotation is usually necessary in supervised learning. However, for most real world problems, it's expensive to obtain high quality training data sets. Learn from imperfect data relies heavily on knowledge distillation, transfer and enhancement. Based on learning methods including transfer learning, metric learning, weakly supervised learning, and etc, in this thesis, we propose novel algorithms for three problems: classifying images with insufficient training data, learning feature embeddings for image distance calculation, and recognizing pixels from image level annotation. Given an image classification task with insufficient training data, selective joint fine- tuning (written as SJFT), is designed to select samples from a source data set which is rich in data annotation to learn the convolutional kernels efficiently. In the context of deep learning, too many learnable parameters and relatively small scale training sets will lead to overfitting, and thus makes learning task specific features very difficult. SJFT aims to improve the generalization ability of the learned deep features by solving two classification tasks in one deep network — one is the source learning task which contains huge amount labeled data, and the other one is target learning task which lacks of training data. Low level characteristics — histogram features generated by Gabor filters or convolutional filters, are used to select valuable samples from a large scale labeled data set. Then this sub set formulates the source learning task and help to increase the

generalization ability of deep models for the target task. Using the low level characteristics, there is less restrictions compared with the high level semantic features during source training set selection. Experimental results indicate adopting the source-target joint training strategy learns stronger discriminative features on ii Caltech 256, MIT Indoor 67, and fine-grained classification problems (Oxford FLowers 102 and Stanford Dogs 120). Deep metric learning or similarity learning Learns a distance function over images with convolutional neural networks. The large scale sampling space and the risk of local optima makes it challenging to design deep metric learning loss functions. In this work, we propose a novel hierarchical triplet loss (written as HTL) able to automatically collect informative training triplets via an adaptively-learned hierarchical class structure that encodes global context in an elegant manner. Specifically, we explore the underline data distribution on a manifold sphere, and then use this manifold structure to guide triplet sample generation. It allows the model to collect informative training samples, alleviates main limitation of random sampling in training of deep metric learning, and encourage the model to learn more discriminative features from visual similar classes. The proposed HTL outperforms the standard triplet loss substantially by 1%-18%, and achieves new state-of-art performance on a number of benchmarks including In-Shop Clothes Retrieval, Caltech-UCSD Birds 200- 2011, Cars 196, and Stanford Online Products. To target the limit of deep neural networks in recognizing pixels from image level labels, we propose a novel weakly supervised curriculum learning pipeline for multi-label object recognition, detection, and semantic segmentation. It is called multi-evidence filtering and fusion (written as MEFF). MEFF follows the divide-and-conquer strategy to solve the weakly supervised learning task with three stages — the pixel level stage, the object level stage, and the pixel level stage. With image level labels, we perform multi-label object recognition at first. Then both metric learning and density-based clustering are incorporated to filter detected object instances. To obtain a relatively clean pixel-wise probability map for every class and every training image, we propose a novel algorithm for fusing image level and object level attention maps with an object detection heat map. Then these intermediate labeling results for the training images are used to train task-specific networks in a fully supervised manner. Experiments show that our weakly supervised pipeline achieves state-of-the-art results in multi-label image classification as well as weakly supervised object detection and very competitive results in weakly supervised semantic segmentation on MS-COCO, PASCAL VOC 2007 and PASCAL VOC 2012. [670 words] Temporary Binding for Examination Purposes Image Understanding from Imperfect Data via Transfer Learning, Metric Learning, and Weakly Supervised Learning Weifeng GE Supervisor: Prof. Yizhou Yu Department of Computer Science The University of Hong Kong This dissertation is submitted for the degree of Doctor of Philosophy Aug 2019 I would like to dedicate this thesis to my beloved family, the one wanted and the dreams to be fulfilled . . . Declaration I declare that this thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in a thesis, dissertation or report submitted to this University or to any other institution for a degree, diploma or other qualifications. Weifeng GE Aug 2019 Acknowledgements First of all, I would like to thank my most respected supervisor, Professor Yizhou Yu, who acts an idol in critical thinking, concrete practice and vibrant creativity. During my Ph.D training, I get full support and supervision from Professor Yu about selecting right problems to solve, finding meaningful points to move related fields forward, and expressing scientific opinions brief and precisely. The experience of working with Professor Yu at HKU will be one of the most important wealth in my life. I sincerely appreciate Xiangru Lin, Sibei Yang, Weilin Huang, and Dengke Dong for their

collaborations in the projects of my thesis. They have provide many constructive suggestions about the algorithm design and the essay writing, and spend a lot of time on the coding works. I want to thank other members in my lab, Guanbin Li, Xiaoguang Han, Wei Zhang, Li Zhang, Zhen Li, Feida Zhu, Chaowei Fang, Haofeng Li, Kan Wu, Bingchen Gong, Sibei Yang, Nenglun Chen, Congyi Zhang and Lei Yang for their accompanies in my long journey to Ph.D. degree. I wish to thank all of my friends and my family who support and encourage me to persevere in my study. Particularly, I am very thankful to my sister Weili Ge for her guidance in my hardest time. Last but not the least, I would like to deliver everlasting feelings of gratefulness and thankfulness to my parents. Finally I wish to express heartfelt thanks to anyone who makes me understand and feel the beauty of the world in different views. Table of contents List of figures xv List of tables xix 1 Introduction 1.1 Research Background and Motivation . . . . . 1. 1.1 Image Classification . . . . . 1.1.2 Image Metric Learning . . . . . 1. 1.3 Weakly Supervised Pixel Recognition . . . . . 1.2 Thesis Outline . . . . . 1.3 Publications . . . . . 1 1 1 3 3 6 6 2 Borrowing Treasures from the Wealthy: Deep Transfer Learning through Se- lective Joint Fine-tuning 2.1 2.2 2.3 2.4 2.5 Introduction . . . . . Related Work . . . . . Selective Joint Fine-tuning . . . . . 2.3.1 Overview . . . . . 2.3.2 Similar Image Search . . . . . Experiments . . . . . 2.4.1 Implementation . . . . . 2.4.2 Source Image Retrieval . . . . . 2.4.3 Fine-grained Object Recognition . . . . . 2.4.4 General Object Recognition . . . . . 2.4.5 Scene Classification . . . . . 2.4.6 Ablation Study . . . . . Conclusions . . . . . 9 9 11 13 13 15 18 18 20 20 22 22 23 23 xii Table of contents 3 Deep Metric Learning with Hierarchical Triplet Loss 3.1 Introduction . . . . . 3.2 Related Work . . . . . 3.3 Motivation: Challenges in Triplet Loss . . . . . 3.3.1 Preliminaries . . . . . 3.3.2 Challenges . . . . . 3.4 Hierarchical Triplet Loss . . . . . 3 .4.1 Manifold Structure in Hierarchy . . . . . 3.4.2 Hierarchical Triplet Loss . . . . . 3.5 Experimental Results and Comparisons . . . . . 3 .5.1 In-Shop Clothes Retrieval . . . . . 3 .5.2 Caltech-UCSD Birds 200-2011 . . . . . 3.5.3 LFW Face Verification . . . . . 3.5.4 Sampling Matter and Local Optima . . . . . 3.5.5 Ablation Study . . . . . 3.6 Conclusions . . . . . 4 27 27 29 30 30 31 32 32 35 37 37 39 40 40 41 42 Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object De- tection and Semantic Segmentation Based on Weakly Supervised Learning 43 4.1 4.2 4.3 Introduction . . . . . Related Work . . . . . Weakly Supervised Curriculum Learning . . . . . 4.3.1 Overview . . . . . 4.3.2 Image Level Stage . . . . . 4.3.3 Instance Level Stage . . . . . 4.3.4 Pixel Level Stage . . . . . 43 45 47 47 48 50 51 4.4 Object Recognition, Detection and Segmentation . . . . . 4. 4.1 Semantic Segmentation . . . . . 4.4.2 Object Detection . . . . . 4.4.3 Multi-label

Performance comparison among multi-label classification methods on Microsoft COCO 2014 validation set. . . . . Comparison among weakly supervised semantic segmentation methods on PASCAL VOC 2012 segmentation val set. . . . . Average precision (in %) of weakly supervised methods on PASCAL VOC 2007 detection test set. . . . . Average precision (in %) of weakly supervised methods on PASCAL VOC 2012 detection test set. . . . . CorLoc (in %) of weakly supervised methods on PASCAL VOC 2007 detection trainval set. . . . . CorLoc (in %) of weakly supervised methods on PASCAL VOC 2012 detection trainval set. . . . . Average precision (in %) of weakly supervised methods on Microsfot COCO 2014 detection validation set. . . . . 60 61 62 63 64 65 66 67 68 Chapter 1 Introduction 1.1 Research Background and Motivation With the breakthrough of AlexNet [56] on ImageNet large scale visual recognition com- petition (ILSRVC) [87], deep convolutional neural networks has dominated the image understanding task in computer vision. However, preparing rich data annotation for deep learning is infeasible and very expensive, and limit the generalization of deep neural networks. This papers focus on three fundamental research topics in computer vision: classifying images with insufficient training data, learning feature embeddings for image distance calculation, and recognizing pixels from image level annotation. To solve the learning task with imperfect data, we design algorithms based transfer learning, metric learning and weakly supervised learning. 1.1.1 Image Classification Deep neural networks require a large amount of labeled training data during supervised learning. However, collecting and labeling so much data might be infeasible in many cases. In this paper, we introduce a source-target selective joint fine-tuning scheme for improving the performance of deep learning tasks with insufficient training data. In this scheme, a target learning task with insufficient training data is carried out simultaneously with another source learning task with abundant training data. However, the source learning task does not use all existing training data. Our core idea is to identify and use a subset of training images from the original source learning task whose low-level characteristics are similar to those from the target learning task, and jointly fine-tune shared convolutional layers for both tasks. Specifically, we compute descriptors from linear or nonlinear filter bank responses on Introduction (a) Source and Target Domain Training Data Source Domain Data Target Domain Data (b) Search k Nearest Neighbors (c) Deep Convolutional (d) Joint Optimization in in Shallow Feature Space Neural Networks Different Label Spaces Source Training Samples Linear classifier in Shallow Feature Spaces Sea Snake Black Grouse … Chameleon Dog Barn Spider Source Domain Loss Minimization Linear classifier Fire Lily Clematis … Rose Artichoke Bee Balm Target Training Samples Convolutional layers shared by the in Shallow Feature Spaces source and target learning tasks Target Domain Loss Minimization Fig. 1. 1 Pipeline of the proposed selective joint fine-tuning. From left to right: (a) Datasets in the source domain and the target domain. (b) Select nearest neighbors of each target domain training sample in the source domain via a low-level feature space. (c) Deep convolutional neural network initialized with weights pre-trained on ImageNet or Places. (d) Jointly optimize the source and target cost functions in their own label spaces. 2 1.1 Research Background and Motivation 3 training images from both tasks, and use such descriptors to search for a desired subset of training samples for the source learning task. 1.1.2 Image Metric Learning We present a novel hierarchical triplet loss (HTL) capable of automatically collecting infor- mative training samples (triplets) via a defined hierarchical tree that encodes global context information. This allows us to cope with the main limitation of random sampling in training a conventional

triplet loss, which is a central issue for deep metric learning. Our main contributions are two-fold. (i) we construct a hierarchical class-level tree where neighboring classes are merged recursively. The hierarchical structure naturally captures the intrinsic data distribution over the whole dataset. (ii) we formulate the problem of triplet collection by introducing a new violate margin, which is computed dynamically based on the designed hierarchical tree. This allows it to automatically select meaningful hard samples with the guide of global context. It encourages the model to learn more discriminative features from visual similar classes, leading to faster convergence and better performance. In addition, the proposed HTL is easily implemented, and the new violate margin can be readily integrated into the standard triplet loss and other deep metric learning functions. 1.1.3 Weakly Supervised Pixel Recognition Supervised object detection and semantic segmentation require object or even pixel level annotations. When there exist image level labels only, it is challenging for weakly supervised algorithms to achieve accurate predictions. The accuracy achieved by top weakly supervised algorithms is still significantly lower than their fully supervised counterparts. In this paper, we propose a novel weakly supervised curriculum learning pipeline for multi-label object recognition, detection and semantic segmentation. In this pipeline, we first obtain intermedi- ate object localization and pixel labeling results for the training images, and then use such results to train task-specific deep networks in a fully supervised manner. The entire process consists of four stages, including object localization in the training images, filtering and fus- ing object instances, pixel labeling for the training images, and task-specific network training. To obtain clean object instances in the training images, we propose a novel algorithm for filtering, fusing and classifying object instances collected from multiple solution mechanisms. In this algorithm, we incorporate both metric learning and density-based clustering to filter detected object instances. 4 Introduction neighbors. (b) Train the convolutional neural network with the hierarchical triplet loss. (c) Online Update of the hierarchical tree. Fig. 1.2 (a)Sampling strategy of each mini-batch. The images in red stand for the anchors and the images in blue stand for the nearest Anchor-Neighbor Groups Conv1 Pool1 Conv2 Pool2 Conv3 Pool3 Conv4 Pool4 Conv5 Pool5 fc Hierarchical Tree Loss on Tree Update Data Distribution in t-SNE Tree Update (a) Anchor Neighbor Sampling (b) Convolutional Neural Network (c) Parameter Optimize and (a) Image Level Stage: Proposal Generation and Multi Evidence Fusion (b) Instance Level Stage: Outlier Detection and (c) Pixel Level Stage: Probability Map Fusion Object Instance Filtering and Pixel Label Prediction Input/Image Object Heatmap Image Attention Map Object Instances Triplet Loss Net Filtered Object Instances Instance Attention Map Instance Classifier Label Map with Uncertainty Probability Map 1.1 Research Background and Motivation Fig. 1.3 The proposed weakly supervised pipeline. From left to right: (a) Image level stage: fuse the object heatmaps H and the image attention map Ag to generate object instances R for the instance level stage, and provide these two maps for information fusion at the pixel level stage. (b) Instance level stage: perform triplet loss based metric learning and density based clustering for outlier detection, and train a single label instance classifier φs(·,·) for instance filtering. (c) Pixel level stage: integrate the object heatmaps H, instance attention map Al, and image attention map Ag for pixel labeling with uncertainty. 5 6 Introduction 1.2 Thesis Outline The reminder of this thesis is arranged as follows. 1) Chapter 2 The proposed selective joint fine-tuning for image classification, Selective Joint Fine-tuning, will be illustrated in Chapter 2. Methodology in fine-tuning frame- works, nearest neighbor searching results and experiments on four datasets (Caltech 256, MIT Indoor 67, Oxford Flowers 102 and Stanford Dogs 120) are introduced. 2) Chapter 3 We present Hierarchical Triplet Loss devised for image retrieval and face

recognition. Challenges in deep metric learning, motivations of hierarchical triplet loss, detailed methodology, and experiments on two image retrieval datasets (In-shot clothes retrieval and Caltech-UCSD Birds 200- 2011) and one face recognition dataset (LFW) are revealed. 3) Chapter 4 The Multi-evidence Fusion and Filtering for multi-label classification, object detection and semantic segmentation is elaborated in Chapter 4. The divide- and-conquer strategy and experiments on multi-label classification task (MS COCO), weakly supervised object detection (PASCAL VOC 2007, PSACAL VOC 2012, and MS COCO), and semantic segmentation (PASCAL VOC 2012) are presented. 4) Chapter 5 In Chapter 5, we summarize the previous methods and discuss the open problems for future research. 1.3 Publications 1 Weakly Supervised Complementary Parts Models for Image Classification from the Bottom Up, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, June 2019. Weifeng Ge∗, Xiangru Lin∗, and Yizhou Yu 2 Image Super-Resolution via Deterministic-Stochastic Synthesis and Local Statistical Rectification, SIGGRAPH Asia 2018, Tokyo, December 2018 (ACM Transcations on Graphics, Vol 37, No 6, 2018). Weifeng Ge∗, Bingchen Gong∗, and Yizhou Yu 3 Deep Metric Learning with Hierarchical Triplet Loss, European Conference on Com- puter Vision (ECCV), Munich, Sep 2018. Weifeng Ge, Weilin Huang, Dengke Dong, and Matthew R. Scott 1.3 Publications 7 4 Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, June 2018. Weifeng Ge, Sibei Yang, and Yizhou Yu 5 Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Fine-Tuning, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawii, July 2017. Weifeng Ge, and Yizhou Yu Chapter 2 Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Fine-tuning 2.1 Introduction Convolutional neural networks (CNNs) have become deeper and larger to pursue increasingly better performance on classification and recognition tasks [56, 49, 102, 39, 37]. Looking at the successes of deep learning in computer vison, we find that a large amount of training or pre-training data is essential in training deep neural networks. Large-scale image datasets, such as the ImageNet ILSVRC dataset [87], Places [131], and MS COCO [62], have led to a series of breakthroughs in visual recognition, including image classification [63], object detection [30], and semantic segmentation [66]. Many other related visual tasks have benefited from these breakthroughs. Nonetheless, researchers face a dilemma when using deep convolutional neural networks to perform visual tasks that do not have sufficient training data. Training a deep network with insufficient data might even give rise to inferior performance in comparison to traditional classifiers fed with handcrafted features. Fine-grained classification problems, such as Oxford Flowers 102 [71] and Stanford Dogs 120 [50], are such examples. The number of training samples in these datasets is far from being enough for training large-scale deep neural networks because a large number of parameters need to be learnt and the networks would become overfit quickly. Solving the overfitting problem for deep convolutional neural networks on learning tasks without sufficient training data is very challenging [99]. Transfer learning techniques that Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint 10 Fine-tuning apply knowledge learnt from one task to other related tasks have been proven helpful [76]. In the context of deep learning, fine-tuning a deep network pre-trained on the ImageNet or Places dataset is a common strategy to learn task-specific deep features [45, 125, 40]. This strategy is considered a simple transfer learning technique for deep learning. However, since the ratio between the number of learnable parameters and the number of training samples still remains the same, fine-tuning needs to

be terminated after a relatively small number of iterations; otherwise, overfitting is still going to occur. In this paper, we attempt to tackle the problem of training deep neural networks for learning tasks that have insufficient training data. We adopt the source-target joint training methodology [122] when fine-tuning deep neural networks. The original learning task without sufficient training data is called the target learning task, T t. To boost its performance, the target learning task is teamed up with another learning task with rich training data. The latter is called the source learning task, T s. Suppose the source learning task has a large-scale training set Ds, and the target learning task has a small-scale training set Dt. Since the target learning task is likely a specialized task, we envisage the image signals in its dataset possess certain unique low-level characteristics (e.g. fur textures in Stanford Dogs 120 [50]), and the learned kernels in the convolutional layers of a deep network need to grasp such characteristics in order to generate highly discriminative features. Thus supplying sufficient training images with similar low-level characteristics becomes the most important mission of the source learning task. Our core idea is to identify a subset of training images from Ds whose low-level characteristics are similar to those from Dt, and then jointly fine-tune a shared set of convolutional layers for both source and target learning tasks. The source learning task is fine-tuned using the selected training images only. Hence, this process is called selective joint fine-tuning. The rationale behind this is that the unique low-level characteristics of the images from Dt might be overwhelmed if all images from Ds were taken as training samples for the source learning task. How do we select images from Ds that share similar low-level characteristics as those from Dt? Since kernels followed with nonlinear activation in a deep convolutional neural network (CNN) are actually nonlinear spatial filters, to find sufficient data for training high- quality kernels, we use the responses from existing linear or nonlinear filter banks to define similarity in low-level characteristics. Gabor filters [70] form an example of a linear filter bank, and the complete set of kernels from certain layers of a pre-trained CNN form an example of a nonlinear filter bank. We use histograms of filter bank responses as image descriptors to search for images with similar low-level characteristics. The motivation behind selecting images according to their low-level characteristics is two fold. First, low-level characteristics are extracted by kernels in the lower convolutional 2.2 Related Work 11 layers of a deep network. These lower convolutional layers form the foundation of an entire network, and the quality of features extracted by these layers determines the quality of features at higher levels of the deep network. Sufficient training images sharing similar low- level characteristics could strength the kernels in these layers. Second, images with similar low-level characteristics could have very different high-level semantic contents. Therefore, searching for images using low-level characteristics has less restrictions and can return much more training images than using high-level semantic contents. The above source-target selective joint fine-tuning scheme is expected to benefit the target learning task in two different ways. First, since convolutional layers are shared between the two learning tasks, the selected training samples for the source learning task prevent the deep network from overfitting quickly. Second, since the selected training samples for the source learning task share similar low-level characteristics as those from the target learning task, kernels in their shared convolutional layers can be trained more robustly to generate highly discriminative features for the target learning task. The proposed source-target selective joint fine-tuning scheme is easy to implement. Ex- perimental results demonstrate state-of-the-art performance on multiple visual classification tasks with much less training samples than what is required by recent deep learning architec- tures. These visual classification tasks include fine-grained classification on

Stanford Dogs 120 [50] and Oxford Flowers 102 [71], image classification on Caltech 256 [33], and scene classification on MIT Indoor 67 [80]. In summary, this paper has the following contributions: • We introduce a source-target selective joint fine-tuning scheme for improving the performance of deep learning tasks with insufficient training data. • We develop a novel pipeline for implementing this selective joint fine-tuning scheme. Specifically, we compute descriptors from linear or nonlinear filter bank responses on training images from both tasks, and use such descriptors to search for a desired subset of training samples for the source learning task. • Experiments demonstrate that our selective joint fine-tuning scheme achieves state-of- the-art performance on multiple visual classification tasks with insufficient training data for deep learning. 2. 2 Related Work Multi-Task Learning. Multi-task learning (MTL) learns shared feature representations or classifiers for related tasks [12]. In comparison to learning individual tasks independently, Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint 12 Fine-tuning features and classifiers learned with MTL often have better generalization capability. The focus of [27] is on learning a shared feature representation that generalizes well on related tasks. Multiple tasks were learned in a joint model in [13] by explicitly optimizing shared parameters and task-specific parameters. In deep learning, faster RCNN [81] jointly learns object locations and labels using shared convolutional layers but different loss functions for these two tasks. In [24], the same multi-scale convolutional architecture was used to predict depth, surface normals and semantic labels. This indicates that convolutional neural networks can be adapted to different tasks easily. While previous work [27, 81] attempts to find a shared feature space that benefits multiple learning tasks, the proposed joint training scheme in this paper focuses on learning a shared feature space that improves the performance of the target learning task only. Feature Extraction and Fine-tuning. Off-the-shelf CNN features [90, 20] have been proven to be powerful in various computer vision problems. Pre-training convolutional neural networks on ImageNet [87] or Places [131] has been the standard practice for other vision problems. However, features learnt in pre-trained models are not tailored for the target learning task. Fine-tuning pre-trained models [30] has become a commonly used method to learn task-specific features. The transfer ability of different convolutional layers in CNNs has been investigated in [125]. However, for tasks that do not have sufficient training data, overfitting occurs quickly. Li et al. [61] proposed to preserve the output for old learning tasks while learning a new task. Their method prevents the network from overfitting quickly on the new task while keeping its discriminative power on the old tasks. The proposed MTL pipeline in this paper not only alleviates overfitting, but also tries to find a more discriminative feature space for the target learning task. Transfer Learning. Different from MTL, transfer learning (or domain adaptation) [76] applies knowledge learnt in one domain to other related tasks. Much work has been done to transfer knowledge between different domains using supervised or unsupervised learning [8, 31, 107]. Domain adaptation algorithms can be divided into three categories, including instance adaption [44, 2], feature adaption [67, 107], and model adaption [21]. Hong et al. [42] transferred rich semantic information from source categories to target categories via the attention model. Tzeng et al. [107] performed feature adaptation using a shared convolutional neural network by transferring the class relationship in the source domain to the target domain. To make our pipeline more flexible, this paper does not assume the source and target label spaces are the same as in [107]. Different from the work in [2] which randomly resamples training classes or images in the source domain, this paper conducts a special type of transfer learning by selecting source training samples that are nearest neighbors of samples in the target

domain in the space of certain low-level image descriptor. 2.3 Selective Joint Fine-tuning 13 Krause et al. [55] directly performed Google image search using keywords associated with categories from the target domain, and download a noisy collection of images to form a training set. Then they apply the classifiers learnt on this training set to images in the target domain. In our method, we search for nearest neighbors in a large-scale labeled dataset using low-level features instead of high-level semantic information. It has been shown in [69] that low-level features computed in the bottom layers of a CNN encode very rich information, which can completely reconstruct the original image. Our experimental results show that nearest neighbor search using low-level features can outperform that using high-level semantic information as in [55]. 2.3 Selective Joint Fine-tuning 2.3.1 Overview Fig. 2.1 shows the overall pipeline for our proposed source-target selective joint fine-tuning scheme. Given a target learning task T t that has insufficient training data, we perform selective joint fine-tuning as follows. The entire training dataset associated with the target learning task is called the target domain. The source domain is defined similarly. Source Domain : The minimum requirement is that the number of images in the source domain, Ds = xsi,ysi i=1, should be large enough to train a deep convolutional neural n s network from s{c(ratch.)I}deally, these training images should present diversified low-level characteristics. That is, running a filter bank on them give rise to as diversified responses as possible. There exist a few large-scale visual recognition datasets that can serve as the source domain, including ImageNet ILSVRC dataset [87], Places [131], and MS COCO [62]. Source Domain Training Images : In our selective joint fine-tuning, we do not use all images in the source domain as training images. Instead, for each image from the target domain, we search a certain number of images with similar low-level characteristics from the source domain. Only images returned from these searches are used as training images for the source learning task in selective joint fine-tuning. We apply a filter bank to all images in both source domain and target domain. Histograms of filter bank responses are used as image descriptors during search. We associate an adaptive number of source domain images with each target domain image. Hard training samples in the target domain might be associated with a larger number of source domain images. Two filter banks are used in our experiments. One is the Gabor filter bank, and the other consists of kernels in the convolutional layers of AlexNet pre-trained on ImageNet [56]. Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Fine-tuning (a) Source and Target Domain Training Data Source Domain Data Target Domain Data 14 (b) Search k Nearest Neighbors (c) Deep Convolutional (d) Joint Optimization in in Shallow Feature Space Neural Networks Different Label Spaces Source Training Samples Linear classifier in Shallow Feature Spaces Sea Snake Black Grouse … Chameleon Dog Barn Spider Source Domain Loss Minimization Linear classifier Fire Lily Clematis … Rose Artichoke Bee Balm Target Training Samples Convolutional layers shared by the in Shallow Feature Spaces source and target learning tasks Target Domain Loss Minimization Fig. 2.1 Pipeline of the proposed selective joint fine-tuning. From left to right: (a) Datasets in the source domain and the target domain. (b) Select nearest neighbors of each target domain training sample in the source domain via a low-level feature space. (c) Deep convolutional neural network initialized with weights pre-trained on ImageNet or Places. (d) Jointly optimize the source and target cost functions in their own label spaces. 2.3 Selective Joint Fine-tuning 15 CNN Architecture : Almost any existing deep convolutional neural network, such as AlexNet [56], GoogleNet [102], VggNet [49], and ResidualNet [37], can be used in our selective joint fine-tuning. We use the 152-layer residual network with identity mappings [39] as the CNN architecture in our experiments. The entire residual network is shared by the source and target

learning tasks. An extra output layer is added on top of the residual network for each of the two learning tasks. This output layer is not shared because the two learning tasks may not share the same label space. The residual network is pre-trained either on ImageNet or Places. Source-Target Joint Fine-tuning : Each task uses its own cost function during selective joint fine-tuning, and every training image only contributes to the cost function corresponding to the domain it comes from. The source domain images selected by the aforementioned searches are used as training images for the source learning task only while the entire target domain is used as the training set for the target learning task only. Since the residual network (with all its convolutional layers) is shared by these two learning tasks, it is fine-tuned by both training sets. And the output layers on top of the residual network are fine-tuned by its corresponding training set only. Thus we conduct end-to-end joint fine-tuning to minimize the original loss functions of the source learning task and the target learning task simultaneously. 2.3.2 Similar Image Search There is a unique step in our pipeline. For each image from the target domain, we search a certain number of images with similar low-level characteristics from the source domain. Only images returned from these searches are used as training images for the source learning task in selective joint fine-tuning. We elaborate this image search step below. Filter Bank We use the responses to a filter bank to describe the low-level characteristics of an image. The first filter bank we use is the Gabor filter bank. Gabor filters are commonly used for feature description, especially texture description [70]. Gabor filter responses are powerful low-level features for image and pattern analysis. We use the parameter setting in [70] as a reference. For each of the real and imaginary parts, we use 24 convolutional kernels with 4 scales and 6 orientations. Thus there are 48 Gabor filters in total. Kernels in a deep convolutional neural network are actually spatial filters. When there is nonlinear activation following a kernel, the combination of the kernel and nonlinear activation is essentially a nonlinear filter. A deep CNN can extract low/middle/high level features at different convolutional layers [125]. Convolutional layers close to the input data focus on extract low-level features while those further away from the input extract middle- and high-level features. In fact, a subset of the kernels in the first convolutional layer of AlexNet 16 Fine-tuning trained on ImageNet exhibit oriented stripes, similar to Gabor filters [56]. When trained on a large-scale diverse dataset, such as ImageNet, such kernels can be used for describing generic low-level image characteristics. In practice, we use all kernels (and their following nonlinear activation) from the first and second convolutional layers of AlexNet pre-trained on ImageNet as our second choice of a filter bank. Fig. 2.2 Convolutional filters and Gabor filters. Image Descriptor Let Ci (m, n) denote the response map to the i-th convolutional kernel or Gabor filter in our filter bank, and φ i its histogram. To obtain more discriminative histogram features, we first obtain the upper bound hui and lower bound hli of the i-th response map by scanning the entire target domain Dt . Then the interval hli, hui is divided into a set of small bins. We adaptively set the width of every histogram bin so that each of them contains a roughly equal percentage of pixels. In this manner, we can avoid a large percentage of pixels falling into the same bin. We concatenate the histograms of all filter response maps to form a feature vector, φ k = φ 1, φ 2, . . . , φ D , for image xk. Nearest Neighbor R{anking Given}the histogram-based descriptor of a training image xit in the target domain, we search for its nearest neighbors in the source domain D s. Note 2.3 Selective Joint Fine-tuning 17 that the number of kernels in different convolutional layers of AlexNet might be different. To ensure equal weighting among different convolutional layers during nearest neighbor search, each histogram of kernel responses is normalized by the total number of kernels in the corresponding layer. Thus the distance between

the descriptor of a source image xsj and that of a target image xti is computed as follows. H xti,xsj = Σ wh[κ(φih,t,φhj,s)+κ(φhj,s,φih,t)], D ( ) h=1 where wh = 1/Nh, Nh is the number of convolutional kernels in the corresponding layer, φih ,t and φhj,s are the h-th histogram for images xti and xsj, and κ(·,·) is the KL-divergence. Hard Samples in the Target Domain The labels of training samples in the target domain have varying degrees of difficulty to satisfy. Intuitively, we would like to seek extra help for those hard training samples in the target domain by searching for more and more nearest neighbors in the source domain. We propose an iterative scheme for this purpose. We calculate the information entropy to measure the classification uncertainty of training samples in the target domain after the m-th iteration as follows. Him = − Σ pmi,clog(pmi,c), C c=1 where C is the number of classes, pmi ,c is the probability that the i-th training sample belongs to the c-th class after a softmax layer in the m-th iteration. Training samples that have high classification uncertainty are considered hard training samples. In the next iteration, we increase the number of nearest neighbors of the hard training samples as in Eq. (2.3.2), and continue fine-tuning the model trained in the current iteration. For a training sample xti in the target domain, the number of its nearest neighbors in the next iteration is defined as follows. Kim+1= ⎧ Kim+σ1, ŷti=yti Kim+σ0, ŷti≠yti and Him ≥ δ (6) ⎪⎪ Kim, ŷti=yti and Him<δ where σ0, σ1 and δ are constan⎪⎩ts, ŷit is predicted label of xti, and Kim is the number of nearest neighbors in the m-th iteration. By changing the number of nearest neighbors for samples in the target domain, the subset of the source domain used as training data evolves over iterations, which in turn gradually changes the feature representation learned in the deep network. In the above equation, we typically set δ = 0.1, σ0 = 4K0 and σ1 = 2K0, where 18 Fine-tuning K0 is the initial number of nearest neighbors for all samples in the target domain. In our experiments, we stop after five iterations. In Table 2.1, we compare the effectiveness of Gabor filters and various combinations of kernels from AlexNet in our selective joint fine-tuning. In this experiment, we use the 50-layer residual network [37] with half of the convolutional kernels in the original architecture. Filter Bank over all Accuracy(%) Conv1-Conv2 in AlexNet Conv1-Conv5 in AlexNet Conv4-Conv5 in AlexNet Gabor Filters Fine-tuning w/o source domain 89.59 88.82 88.48 88.90 88.12 Table 2 .1 A comparison of classification performance on Oxford Flowers 102 using various choices for the filter bank in selective joint fine-tuning. 2.4 Experiments 2 .4.1 Implementation In all experiments, we use the 152-layer residual network [37] as the deep convolutional architecture. To use GPU memory more efficiently, we modify the implementation of the 152- layer residual network with identity mappings [39] such that more images can be included in a mini-batch using Caffe [47]. We use the pre-trained model released in [37] to initialize the residual network, and choose either ImageNet or the combination of ImageNet and Places as the source domain. During selective joint fine-turning, source and target samples are mixed together in each mini-batch. Once the data has passed the average pooling layer in the residual network, we split the source and target samples, and send them to their corresponding softmax classifier layer respectively. Both the source and target classifiers are initialized randomly. We run all our experiments on a TITAN X GPU with 12GB memory. All training data is augmented as in [78] first, and we follow the training and testing settings in [37]. Every mini-batch can include 20 224×224 images using the reimplemented residual network. We include randomly chosen samples from the target domain in a mini-batch. Then for each of the chosen target sample, we further include one of its retrieved nearest neighbors from the source domain in the same mini-batch. We set the iter size to 10 for each iteration in Caffe a.1 Chihuahua a.2 Chihuahua a.3 Beagle a.4 Tench a.5 Bib a.5 Diaper b.1 Pink Primrose b.2 Bee b.3 Capuchin b.4 Measuring Cup b.4

Butterfly b.5 Bee 2.4 Experiments c.1 AK-47 c.2 Horse Cart c.3 Horn c.4 Hard Disk c.5 Snowmobile c.6 Jeep d.1 Airport Inside d.2 Restaurant d.3 Restaurant d.4 Butcher Shop d.5 Coffee Mug d.6 Grocery Store e.1 Airport Inside e.2 Game Room e.3 Restaurant e.4 Supermarket e.5 Lobby e.5 Museum Fig. 2. 3 Images in the source domain that have similar low-level characteristics with the target images. The first column shows target images from Stanford Dogs 120 [50], Oxford Flowers 102 [71], Caltech 256 [33], and MIT Indoor 67 [80]. The following columns in rows (a)-(d) are the corresponding 1st, 10-th, 20-th, 30-th and 40-th nearest images in ImageNet (source domain). The following columns in row (e) are images retrieved from Places (source domain for MIT Indoor 67). 19 20 Fine-tuning [47]. The momentum parameter is set to 0.9 and the weight decay is 0.0001 in SGD. During selective joint fine-tuning, the learning rate starts from 0.01 and is divided by 10 after every 2400 − 5000 iterations in all the experiments. Most of the experiments can finish in 16000 iterations. 2.4.2 Source Image Retrieval We use the ImageNet ILSVRC 2012 training set [87] as the source domain for Stanford Dogs [50], Oxford Flowers [71], and Caltech 256 [33], and the combination of the ImageNet and Places 205 [131] training sets as the source domain for MIT Indoor 67 [80]. Fig. 2. 3 shows the retrieved 1-st, 10-th, 20-th, 30-th, and 40-th nearest neighbors from ImageNet [87] or Places [131]. It can be observed that corresponding source and target images share similar colors, local patterns and global structures. Since low-level filter bank responses do not encode strong semantic information, the 50 nearest neighbors from a target domain include images from various and sometimes completely unrelated categories. We find out experimentally that there should be at least 200,000 retrieved images from the source domain. Too few source images give rise to overfitting quickly. Therefore, the initial number of retrieved nearest neighbors (K0) for each target training sample is set to meet this requirement. On the other hand, a surprising result is that setting K0 too large would make the performance of the target learning task drop significantly. In our experiments, we set K0 to different values for Stanford Dogs (K0 = 100), Oxford Flowers (K0 = 300), Caltech 256 (K0 = 50 − 100), and MIT Indoor 67 (K0 = 100). Since there exists much overlap among the nearest neighbors of different target samples, the retrieved images typically do not cover the entire ImageNet or Places datasets. 2.4.3 Fine-grained Object Recognition Stanford Dogs 120. Stanford Dogs 120 [50] contains 120 categories of dogs. There are 12000 images for training, and 8580 images for testing. We do not use the parts information during selective joint fine-tuning, and use the commonly used mean class accuracy to evaluate the performance as in [33]. As shown in Table 2. 2, the mean class accuracy achieved by fine-tuning the residual network using the training samples of this dataset only and without a source domain is 80.4%, which is already better than most previous state-of-the-art results except [55]. It shows that the 152-layer residual network [37, 39] pre-trained on the ImageNet dataset [87] has a strong generalization capability on this fine-grained classification task. Using the entire ImageNet dataset during regular joint fine-tuning can improve the performance by 5.1%. When we fi- 2.4 Experiments 21 Method mean Acc(%) HAR-CNN [121] Local Alignment [28] Multi scale metric learning [79] MagNet [83] Web Data + Original Data [55] 49.4 57.0 70.3 75.1 85.9 Fine-tuning w/o source domain Selective joint FT with all source sample Selective joint FT with random source samples Selective joint FT w/o iterative NN retrieval Selective joint FT with Gabor filter bank Selective joint FT Selective joint FT with Model Fusion 80.4 85.6 85.5 88.3 87.5 90.2 90.3 Table 2. 2 Classification results on Stanford Dogs 120. nally perform our proposed selective joint fine-tuning using a subset of source domain images retrieved using histograms of low-level convolutional features, the performance is further improved to 90.2%, which is 9.8%

higher than the performance of conventional fine-tuning without a source domain and 4.3% higher than the result reported in [55], which expands the original target training set using Google image search. This comparison demonstrates that selective joint fine-tuning can significantly outperform conventional fine-tuning. Oxford Flowers 102. Oxford Flowers 102 [71] consists of 102 flower categories. 1020 images are used for training, 1020 for validation, and 6149 images are used for testing. There are only 10 training images in each category. As shown in Table 2.3, the mean class accuracy achieved by conventional fine-tuning using the training samples of this dataset only and without a source domain is 92.3%. Selective joint fine-tuning further improves the performance to 94.7%, 3.3% higher than previous best result from a single network [83]. To compare with previous state-of-the-art results obtained using an ensemble of different networks, we also average the performance of multiple models obtained during iterative source image retrieval for hard training samples in the target domain. Experiments show that the performance of our ensemble model is 95.8%, 1.3% higher than previous best ensemble performance reported in [52]. Note that Simon et al. [94] used the validation set in this dataset as additional training data. To verify the effectiveness of our joint fine-tuning strategy, we have also conducted experiments with this setting during training and our result from a single network outperforms that of [94] by 1.7%. 22 Fine-tuning 2 .4.4 General Object Recognition Caltech 256. Caltech 256 [33] has 256 object categories and 1 background cluster class. In every category, there are at least 80 images used for training, validation and testing. Researchers typically report results with the number of training samples per class falling between 5 and 60. We follow the testing procedure in [113] to compare with state-of-the-art results. We conduct four experiments with the number of training samples per class set to 15, 30, 45 and 60, respectively. According to Table 2.4, in comparison to conventional fine-tuning without using a source domain, selective joint fine-tuning improves classification accuracy in all four experiments, and the degree of improvement varies between 2.6% and 4.1%. Performance improvement due to selective joint fine-tuning is more obvious when a smaller number of target training image per class are used. This is because limited diversity in the target training data imposes a greater need to seek help from the source domain. In most of these experiments, the classification performance of our selective joint fine-tuning is also significantly better than previous state-of-the-art results. 2 .4.5 Scene Classification MIT Indoor 67. MIT Indoor 67 [80] has 67 scene categories. In each category, there are 80 images for training and 20 images for testing. Since MIT Indoor 67 is a scene dataset, in addition to the ImageNet ILSVRC 2012 training set [87], the Places-205 training set [131] is also a potential source domain. We compare three settings during slective joint fine-tuning: ImageNet as the source domain, Places as the source domain, and the combination of both ImageNet and Places as the source domain. As shown in Table 2.5, the mean class accuracy of selective joint fine-tuning with ImageNet as the source domain is 82.8%, 1.1% higher than that of conventional fine-tuning without using a source domain. Since ImageNet is an object-centric dataset while MIT Indoor 67 is a scene dataset, it is hard for training images in the target domain to retrieve source domain images with similar low-level characteristics. But source images retrieved from ImageNet still prevent the network from overfitting too heavily and help achieve a performance gain. When the Places dataset serves as the source domain, the mean class accuracy reaches 85.8%, which is 4.1% higher than the performance of fine-tuning without a source domain and 4.8% higher than previous best result from a single network [16]. And the hybrid source domain based on both ImageNet and Places does not further improve the performance. Once averaging the output from the networks jointly fine-tuned

with Places 2.5 Conclusions 23 and the hybrid source domain, we obtain a classification accuracy 0.9% higher than previous best result from an ensemble model [40]. 2 .4.6 Ablation Study We perform an ablation study on both Stanford Dogs 120 [50] and Oxford Flowers 102 [71] by replacing or removing a single component from our pipeline. First, instead of using a subset of retrieved training images from the source domain, we simply use all training images in the source domain. Table 2.2 and Table 2.3 show that joint fine-tuning with the entire source domain decrease the performance by 4.6% and 1.3% respectively. This demonstrates that using more training data from the source domain is not always better. On the contrary, using less but more relevant data from the source domain is actually more helpful. Second, instead of using a subset of retrieved training images, we use the same number of randomly chosen training images from the source domain. Again, the performance drops by 4.7% and 1.5% respectively. Third, to validate the effectiveness of iteratively increasing the number of retrieved images for hard training samples in the target domain, we turn off this feature and only use the same number (K0) of retrieved images for all training samples in the target domain. The performance drops by 1.9% and 0.5% respectively. This indicates that our adaptive scheme for hard samples is useful in improving the performance. Fourth, we use convolutional kernels in the two bottom layers of a pre-trained AlexNet as our filter bank. If we replace this filter bank with the Gabor filter bank, the overall performance drops by 2.7% and 0.9% respectively, which indicates a filter bank learned from a diverse dataset could be more powerful than an analytically defined one. Finally, if we perform conventional fine-tuning without using a source domain, the performance drop becomes quite significant and reaches 9.8% and 2.4% respectively. 2.5 Conclusions In this paper, we address deep learning tasks with insufficient training data by introducing selective joint fine-tuning, which performs a target learning task with insufficient training data simultaneously with another source learning task with abundant training data. Different from previous work which directly adds extra training data to the target learning task, we try to reduce the amount of data annotation effort when solving visual recognition problems. We borrow samples from a large-scale labeled database such as ImageNet and Places, and do not require additional labeling effort beyond the existing datasets. Experiments show that our selective joint fine-tuning strategy achieves state-of-the-art performance on multiple visual 24 Fine-tuning classification tasks with insufficient training data for deep learning. Nevertheless, how to find the most suitable source domain for a specific target learning task remains an open problem for future investigation. 2.5 Conclusions 25 Method mean Acc(%) MPP [124] Multi-model feature contact [2] MagNet [83] VGG-19 + GoogleNet + AlexNet [52] 91.3 91.3 91.4 94.5 Fine-tuning w/o source domain Selective joint FT with all source samples Selective joint FT with random source samples Selective joint FT w/o iterative NN retrieval Selective joint FT with Gabor filter bank Selective joint FT Selective joint FT with model fusion 92 .3 93.4 93.2 94.2 93.8 94.7 95.8 VGG-19 + Part Constellation Model [94] 95.3 Selective joint FT with val set 97.0 Table 2.3 Classification results on Oxford Flowers 102. The last two rows compare perfor- mance using the validation set as additional training data. #Train mean Acc(%) mean Acc(%) mean Acc(%) mean Acc(%) 15/class 30/class 45/class 60/class M-HMP [10] Z. & F. Net [127] VGG-19 [49] VGG-19 + GoogleNet +AlexNet [52] VGG-19 + VGG-16 [49] 40.5 ±0. 4 65.7±0.2 - - - 48.0±0.2 70.6±0.2 - - - 51.9±0.2 72.7±0.4 - - - 55.2±0.3 74.2±0.3 85.1±0.3 86.1 86.2±0.3 Fine-tuning w/o source domain 76.4±0.1 81.2±0.2 83.5±0.2 86.4±0.3 Selective joint FT 80.5±0.3 83.8±0.5 87.0±0.1 89.1±0.2 Table 2.4 Classification results on Caltech 256. 26 Fine-tuning Method mean Acc(%) MetaObject-CNN [120] MPP + DFSL [124] VGG-19 + FV [16] VGG-19 + GoogleNet [52] Multi scale + multi model ensemble [40] 78.9 80.8

81.0 84.7 86.0 Fine-tuning w/o source domain Selective joint FT with ImageNet(i) Selective joint FT with Places(ii) Selective joint FT with hybrid data(iii) Average the output of (ii) and (iii) 81.7 82.8 85.8 85.5 86.9 Table 2 .5 Classification results on MIT Indoor 67. Chapter 3 Deep Metric Learning with Hierarchical Triplet Loss 3. 1 Introduction Distance metric learning or similarity learning is the task of learning a distance function over images in visual understanding tasks. It has been an active research topic in computer vision community. Given a similarity function, images with similar content are projected onto neighboring locations on a manifold, and images with different semantic context are mapped apart from each other. With the boom of deep neural networks (DNN), metric learning has been turned from learning distance functions to learning deep feature embeddings that better fits a simple distance function, such as Euclidean distance or cosine distance. Metric learning with DNNs is referred as deep metric learning, which has recently achieved great success in numerous visual understanding tasks, including images or object retrieval [98, 109, 117], single-shot object classification [109, 117, 112], keypoint descriptor learning [57, 93], face verification [89, 77], person re-identification [109, 91], object tracking [105] and etc. Recently, there is a number of widely-used loss functions developed for deep metric learning, such as contrastive loss [101, 34], triplet loss [89] and quadruplet loss [15]. These loss functions are calculated on correlated samples, with a common goal of encouraging samples from the same class to be closer, and pushing samples of different classes apart from each other, in a projected feature space. The correlated samples are grouped into contrastive pairs, triplets or quadruplets, which form the training samples for these loss functions on deep metric learning. Unlike softmax loss used for image classification, where the gradient is computed on each individual sample, the gradient of a deep metric learning loss often depends heavily on multiple correlated samples. Furthermore, the number of training samples will be 28 Deep Metric Learning with Hierarchical Triplet Loss increased exponentially when the training pairs, triplets or quadruplets are grouped. This generates a vast number of training samples which are highly redundant and less informative. Training that uses random sampling from them can be overwhelmed by redundant samples, leading to slow convergence and inferior performance. Deep neural networks are commonly trained using online stochastic gradient descent algorithms [75], where the gradients for optimizing network parameters are computed locally with mini-batches, due to the limitation of computational power and memory storage. It is difficult or impossible to put all training samples into a single mini-batch, and the networks can only focus on local data distribution within a mini-batch, making it difficult to consider global data distribution over the whole training set. This often leads to local optima and slow convergence. This common challenge will be amplified substantially in deep metric learning, due to the enlarged sample spaces where the redundancy could become more significant. Therefore, collecting and creating meaningful training samples (e.g., in pairs, triplets or quadruplets) has been a central issue for deep metric learning, and an efficient sampling strategy is of critical importance to this task. This is also indicated in recent literature [89, 118, 77, 1]. Our goal of this paper is to address the sampling issue of conventional triplet loss [89]. In this work, we propose a novel hierarchical triplet loss able to automatically collect informative training triplets via an adaptively-learned hierarchical class structure that encodes global context in an elegant manner. Specifically, we explore the underline data distribution on a manifold sphere, and then use this manifold structure to guide triplet sample generation. Our intuition of generating meaningful samples is to encourage the training samples within a mini-batch to have similar visual appearance but with different semantic content (e.g., from different categories). This allows our model to learn more

discriminative features by identifying subtle distinction between the close visual concepts. Our main contribution are described as follows. — We propose a novel hierarchical triplet loss that allows the model to collect informative training samples with the guide of a global class-level hierarchical tree. This alleviates main limitation of random sampling in training of deep metric learning, and encourages the model to learn more discriminative features from visual similar classes. — We formulate the problem of triplet collection by introducing a new violate margin, which is computed dynamically over the constructed hierarchical tree. The new violate margin allows us to search informative samples, which are hard to distinguish between visual similar classes, and will be merged into a new class in next level of the hierarchy. The violate margin is automatically updated, with the goal of identifying a margin that generates gradients for violate triplets, naturally making the collected samples more informative. 3.2 Related Work 29 — The proposed hierarchical triplet loss is easily implemented, and can be readily integrated into the standard triplet loss or other deep metric learning approaches, such as contrastive loss, quadruplet loss, recent HDC [126] and BIER [72]. It significantly outperforms the standard triplet loss on the tasks of image retrieval and face recognition, and obtains new state-of-art results on the CUB-200-2011 [111], and In-Shop Clothes Retrieval [65]. 3 .2 Related Work Deep Metric Learning. Deep metric learning maps an image into a feature vector in a manifold space via deep neural networks. In this manifold space, the Euclidean distance (or the cosin distance) can be directly used as the distance metric between two points. The contribution of many deep metric learning algorithms, such as [98, 89, 15, 4, 5], is the design of a loss function that can learn more discriminant features. Since neural networks are usually trained using the stochastic gradient descent (SGD) in mini-batches, these loss functions are difficult to approximate the target of metric learning - pull samples with the same label into nearby points and push samples with different labels apart. Informative Sample Selection. Given N training images, there are about O(N2) pairs, O(N3) triplets, and 0(N4) quadruplets. It' s infeasible to traverses all these training tuples during training. Schroff at. el [89] constructed a mini-batch of with 45 identities and each of which has 40 images. There are totally 1800 images in a mini-batch, and the approach obtained the state-of-art results on LFW face recognition challenge [43]. While it is rather inconvenient to take thousands of images in a mini-batch with a large-scale network, due to the limitation of GPU memory. For deep metric learning, it is of great importance to selecting informative training tuples by ignoring or discarding the easy samples. Hard nega- tive mining [11] is widely used to select hard training tuples. Our work is closely related to that of [118, 36] that inspired the current work. The authors of [118, 36] applied distance distribution to guide tuple sampling for deep metric learning. In this work, we strive to a further step by constructing a hierarchical tree that aggregates class-level global context, and formulating tuple selection elegantly by introducing a new violate margin. 30 Deep Metric Learning with Hierarchical Triplet Loss (a) Caltech-UCSD Bird Species Dataset (b) Data Distribution and Triplets in a Mini-Batch Fig. 3.1 (a) Caltech-UCSD Bird Species Dataset [111]. Images in each row are from the same class. There are four classes in different colors — red, green, blue and yellow. (b) Data distribution and triplets in a mini-batch. Triplets in the top row violate the triplet constrain in the traditional triplet loss. Triplets in the bottom row are ignored in the triplet loss, but are revisited in the hierarchical triplet loss. 3. 3 Motivation: Challenges in Triplet Loss We start by revisiting the main challenges in standard triplet loss [89], which we believe have a significant impact to the performance of deep triplet embedding. We study and discuss insights of these problems that inspire current work. 3.3.1 Preliminaries Let (xi,yi) be the i-th sample in the training set D = {(xi,yi)}iN=1. The feature

embedding of xi is represented as φ (xi,θ) ∈ Rd, where θ is the learnable parameters of a differentiable deep networks, d is the dimension of embedding and yi is the label of xi. φ (·,θ) is usually normalized into unit length for the training stability and comparison simplicity as in [89]. During the neural network training, training samples are selected and formed into triplets, each of which Tz = (xa,xp,xn) are consisted of an anchor sample xa, a positive sample xp and a negative sample xn. The labels of the triplet Tz = xza,xzp,xzn satisfy ya = yp ≠ yn. Triplet loss aims to pull samples belonging to the same class(into nearb)y points on a manifold surface, and push samples with different labels apart from each other. The optimization target of the triplet Tz is, ltri (Tz) = xza − xzp 2 − ‖xza − xzn‖2 + α 1 2 . [ ] + ‖ ‖ ‖ 3 .3 Motivation: Challenges in Triplet Loss 31 [·]+ = max(0, ·) denotes the hinge loss function, and α is the violate margin that requires the distance ‖xza − xzn ‖2 of negative pairs to be larger than the distance xza − xzp of positive 2 pairs. For all the triplets T in the training set D = {(xi, yi)}iN=1, the fi‖nal objective function to optimize is, ‖ L = Σ ltri(Tz), 1 Z Tz ∈T where Z is the normalization term. For training a triplet loss in deep metric learning, the violate margin plays a key role to sample selection. 3 .3.2 Challenges Challenge 1: triplet loss with random sampling. For many deep metric learning loss functions, such as contrastive loss [34], triplet loss [89] and quadruplet loss [15], all training samples are treated equally with a constant violate margin, which only allows training samples that violate this margin to produce gradients. For a training set D = {(xi, yi)}kN =1 with N samples, training a triplet loss will generate O N3 triplets, which is infeasible to put all triplets into a single mini-batch. When we sampl(e the) triplets over the whole training set randomly, it has a risk of slow convergence and pool local optima. We identify the problem that most of training samples obey the violate margin when the model starts to converge. These samples can not contribute gradients to the learning process, and thus are less informative, but can dominate the training process, which significantly degrades the model capability, with a slow convergence. This inspired current work that formulates the problem of sample selection via setting a dynamic violate margin, which allows the model to focus on a small set of informative samples. However, identifying informative samples from a vast number of the generated triplets is still challenging. This inspires us to strive to a further step, by sampling meaningful triplets from a structural class tree, which defines class-level relations over all categories. This transforms the problem of pushing hard samples apart from each other into encouraging a larger distance between two confusing classes. This not only reduces the search space, but also avoid over-fitting the model over individual samples, leading to a more discriminative model that generalizes better. Challenge 2: risk of local optima. Most of the popular metric learning algorithms, such as the contrastive loss, the triplet loss, and the quadruplet loss, describe similarity rela- tionship between individual samples locally in a mini-batch, without considering global data distribution. In triplet loss, all triplet is treated equally. As shown in Fig. 3.1, when 32 Deep Metric Learning with Hierarchical Triplet Loss the training goes after several epoches, most of training triplets dose not contribute to the gradients of learnable parameters in deep neural networks. There has been recent work that aims to solve this problem by re-weighting the training samples, as in [119]. However, even with hard negative mining or re-weighting, the triplets can only see a few samples within a mini-batch, but not the whole data distribution. It is difficult for the triplet loss to incorporate the global data distribution on the target manifold space. Although the data structure in the deep feature space are changed dynamically during the training process, the relative position of data points can be roughly preserved. This allows us to explore the data distribution obtained in the previous iterations to guide sample selection in

the current stage. With this prior knowledge of data structure, a triplet, which does not violate the original margin α , is possible to generate gradients that contribute to the network training, as shown in Fig. 3.1. Discriminative capability can be enhanced by learning from these hard but informative triplets. 3.4 Hierarchical Triplet Loss We describe details of the proposed hierarchical triplet loss, which contains two main components, constructing a hierarchical class tree and formulating the hierarchical triplet loss with a new violate margin. The hierarchical class tree is designed to capture global data context, which is encoded into triplet sampling via the new violate margin, by formulating the hierarchical triplet loss. 3 .4.1 Manifold Structure in Hierarchy We construct a global hierarchy at the class level. Given a neural network φt (·,θ)(∈ Rd) pre-trained using the traditional triplet loss, we get the hierarchical data structure based on sample rules. Denote the deep feature of a sample xi as ri = φt (xi ,θ). We first calculate a distance matrix of C classes in the whole training set D. The distance between the p-th class and the q-th class is computed as, d(p,q)= npnqi ∈∑p, j ∈q ri−rj 1 2 , where np and nq are the numbers of training sam‖ples in the p-th and the q-th classes ‖ respectively. Since the deep feature ri is normalized into unit length, the value of the interclassdistanced (p,q) variesfrom0to4. 3.4 Hierarchical Triplet Loss 33 (a) Hierarchical Tree (c) Data Distribution Visualization by t-SNE Fig. 3.2 (a) A toy example of the hierarchical tree H . Different colors represent different image classes in CUB-200-2011 [111]. The leaves are the image classes in the training set. Then they are merged recursively until to the root node. (b) The training data distribution of 100 classes visualized by using t-SNE [68] to reduce the dimension of triplet embedding from 512 to 2. We build hierarchical manifold structure by creating a hierarchical tree, according to the computed interclass distances. The leaves of the hierarchical tree are the original image classes, where each class represents a leave node at the 0-th level. Then hierarchy is created by recursively merging the leave notes at different levels, based on the computed distance matrix. The hierarchical tree is set into L levels, and the average inner distance d0 is used as the threshold for merging the nodes at the 0-th level. d0 = C c∑=1 (n2c − nc i∈∑c,j∈c ri − r j 1 C 1 2) . where nc is the number of samples in the c-th clas‖s. Then the nodes are merged with ‖ different thresholds. At the l-th level of the hierarchical tree, the merging threshold is set to dl = l(4 −d0) + d0. Two classes with a distance less than dl are merged into a node at the L l -th level. The node number at the l-th level is Nl. The nodes are merged from the 0-th level to the L-th level. Finally, we generate a hierarchical tree H which starts from the leave nodes of original image classes to a final top node, as shown in Fig. 3.2 (a). The constructed hierarchical tree captures class relationships over the whole dataset, and it is updated interactively at the certain iterations over the training. 34 Deep Metric Learning with Hierarchical Triplet Loss neighbors. (b) Train the convolutional neural network with the hierarchical triplet loss. (c) Online Update of the hierarchical tree. Fig. 3. 3 (a)Sampling strategy of each mini-batch. The images in red stand for the anchors and the images in blue stand for the nearest Anchor-Neighbor Groups Conv1 Pool1 Conv2 Pool2 Conv3 Pool3 Conv4 Pool4 Conv5 Pool5 fc Hierarchical Tree Loss on Tree Tree Update Update Data Distribution in t-SNE (a) Anchor Neighbor Sampling (b) Convolutional Neural Network (c) Parameter Optimize and 3.4 Hierarchical Triplet Loss 35 3 .4.2 Hierarchical Triplet Loss We formulate the problem of triplet collection into a hierarchical triplet loss. We introduce a dynamical violate margin, which is the main difference from the conventional triplet loss using a constant violate margin. Anchor neighbor sampling. We randomly select l′ nodes at the 0-th level of the constructed hierarchical tree H . Each node represents an original class, and collecting classes at the 0-th level aims to preserve the diversity of training samples in a mini-batch, which is important for training deep

networks with batch normalization [46]. Then m − 1 nearest classes at the 0-th level are selected for each of the l′ nodes, based on the distance between classes computed in the feature space. The goal of collecting nearest classes is to encourage model to learn discriminative features from the visual similar classes. Finally, t images for each class are randomly collected, resulting in n (n = l′mt) images in a mini-batch M . Training triplets within each mini-batch are generated from the collected n images based on class relationships. We write the anchor-neighbor sampling into A-N sampling for convenience. Triplet generation and dynamic violate margin. Hierarchical triplet loss (computed on a mini-batch of M) can be formulated as, LM = 2ZM T z∑∈T xza − xzp − ‖xza − xnz‖ + αz 1 M + . where T M is all the triplets in the min[i -‖‖batch M‖, and ZM = A2l′mAt2Ct1 is the number of] triplets. Each triplet is constructed as Tz = (xa,xp,xn), and the training triplets are generated as follows. A2l ′m indicates randomly selecting two classes - a positive class and a negative class, from all l′m classes in the mini-batch. A2t means selecting two samples - a anchor sample (xza) and a positive sample (xzp), from the positive class, and Ct1 means randomly selecting a negative sample (xzn) from the negative class. A2l ′m, At2 and Ct1 are notations in combinatorial mathematics. See reference [110] for details. αz is a dynamic violate margin, which is different from the constant margin of traditional triplet loss. It is computed according to the class relationship between the anchor class ya and the negative class yn over the constructed hieratical class tree. Specifically, for a triplet Tz, the violate margin αz is computed as, αz = β + dH (ya, yn) − sya, 36 Deep Metric Learning with Hierarchical Triplet Loss where β (= 0.1) is a constant parameter that encourages the image classes to reside further apart from each other than the previous iterations. H (ya, yn) is the hierarchi- cal level on the class tree, where the class ya and the class yn are merged into a single node in the next level. dH (ya, yn) is the threshold for merging the two classes on H , and sya = n2ya −1nya ∑i, j∈ya ri − r j 2 is the average distance between samples in the class ya. In our hierarchical triplet lo‖ss, a sa‖mple xa is encouraged to push the nearby points with different semantic meanings a‖part fro‖m itself. Furthermore, it also contributes to the gradients of data points which are very far from it, by computing a dynamic violate margin which encodes global class structure via H . For every individual triplet, we search on H to encode the context information of the data distribution for the optimization objective. Details of training process with the proposed hierarchical triplet loss are described in Algorithm 1. Algorithm 1: Training with hierarchical triplet loss Input: Training data D = {(xi, yi)}kN =1. Network φ (·, θ) is initialized with a pretrained ImageNet model. The hierarchical class tree H is built according to the features of the initialized model. The margin αz for any pair of classes is set to 0.2 at the beginning. Output: The learnable parameters θ of the neural network φ (·, θ). 1 while not converge do 2 t ← t + 1 ; 3 Sample anchors randomly and their neighborhoods according to H ; 4 Compute the violate margin for different pairs of image classes by searching through the hierarchical tree H ; 5 Compute the hierarchical triplet loss in a mini-batch LM ; 6 Backpropagate the gradients produced at the loss layer and update the learnable parameters ; 7 At each epoch, update the hierarchical tree H with current model. Implementation Details All our experiments are implemented using Caffe [47] and run on an NVIDIA TITAN X(Maxwell) GPU with 12GB memory. The network architecture is a GoogLeNet [103] with batch normalization [46] which is pre-trained on the ImageNet dataset [86]. The 1000-way fully connected layer is removed, and replace by a d dimensional fully connected layer. The new added layer is initialized with random noise using the "Xaiver" filler. We modify the memory management of Caffe [47] to ensure it can take 650 images in a mini-batch for GoogLeNet with batch normalization. The

input images are resized and cropped into 3.5 Experimental Results and Comparisons 37 R@ 1 10 20 30 40 50 FashionNet+Joints[65] FashionNet+Poselets[65] FashionNet[65] HDC[126] BIER[73] 41.0 64.0 42.0 65.0 53.0 73.0 62.1 84.9 76.9 92.8 68.0 71.0 70.0 72.0 76.0 77.0 89.0 91.2 95.2 96.2 73.0 72.0 79.0 92.3 96.7 73.5 75.0 80.0 93.1 97.1 Ours Baseline A-N Sampling HTL 62.3 75.3 80.9 85.1 89.0 91.8 94.3 94.3 95.8 91.1 92.4 96.2 96.7 97.2 97.4 93.4 97.5 97.8 Table 3. 1 Comparison with the state-of-art on the In-Shop Clothes Retrieval dataset [65]. 224 × 224, and then subtract the mean value. The optimization method used is the standard SGD with a learning rate 1e−3. 3 .5 Experimental Results and Comparisons We evaluate the proposed hierarchical triplet loss on the tasks of image retrieval and face recognition. Extensive experiments are conducted on a number of benchmarks, including In-Shop Clothes Retrieval [65] and Caltech-UCSD Birds 200 [111] for image retrieval, and LFW [43] for face verification. Descriptions of dataset and implementation details are presented as follows. 3 .5.1 In-Shop Clothes Retrieval Datasets and performance measures. The In-Shop Clothes Retrieval dataset [65] is very popular in image retrieval. It has 11735 classes of clothing items and 54642 training images. Following the protocol in [65, 126], 3997 classes are used for training (25882 images) and 3985 classes are for testing (28760 images). The test set are partitioned into the query set and the gallery set, both of which has 3985 classes. The query set has 14218 images and the gallery set has 12612 images. As in Fig. 3.4, there are a lot image classes that have very similar contents. For the evaluation, we use the most common Recall@K metric. We extract the features of each query image and search the K most similar images in the gallery set. If one of the K retrieved images have the same label with the query image, the recall will increase by 1, otherwise will be 0. We evaluate the recall metrics with K ∈ {1, 2, 4, 8, 16, 32}. Implementation details. Our network is based on GoogLeNet V2 [46]. The dimension d of the feature embedding is 128. The triplet violate margin is set to 0.2. The hierarchical tree 38 Deep Metric Learning with Hierarchical Triplet Loss Fig. 3.4 Anchor-Neighbor visualization on In-Shop Clothes Retrieval training set [65]. Each row stands for a kind of fashion style. The row below each odd row is one of neighborhoods of the fashion style in the odd row. has 16 levels including the leaves level which contains the images classes. At the first epoch, the neural network is trained with the standard triplet loss which samples image classes for mini-batches randomly. Then during the training going on, the hierarchical tree is updated and used in the following steps. Since there are 3997 image classes for training and there many similar classes, the whole training needs 30 epoch and the batch size is set to 480. For every 10 epoch, we decrease the learning rate by multiplying 0.1. The testing codes are gotten from HDC [126]. Result comparison. We compare our method with existing state-of-the-art algorithms and our baseline — triplet loss. Table 3.1 lists the results of image retrieval on In-Shop Clothes Retrieval. The proposed method achieves 80.9% Recall@1, and outperforms the baseline algorithm — triplet loss by 18.6%. It indicates that our algorithm can improve the discriminative power of the original triplet loss by a large margin. State-of-the-art algorithms, including HDC [126], and BIER [73], used boosting and ensemble method to take the advantage of different features and get excellent results. Our method demonstrates that by incorporate the global data distribution into deep metric learning, the performance will be 3.5 Experimental Results and Comparisons 39 R@ 1 2 4 8 16 32 LiftedStruct[98] Binomial Deviance[109] Histogram Loss[109] N-Pair-Loss[97] HDC[126] BIER[73] 47.2 58.9 52.8 64.4 50.3 61.9 51.0 63.3 53.6 65.7 55.3 67.2 70.2 80.2 74.7 83.9 72.6 82.4 74.3 83.2 77.0 85.6 76.9 85.1 89.3 90.4 88.8 - 91.5 91.7 93.2 94.3 93.7 - 95.5 95.5 Ours Baseline 55.9 68.4 78.2 86.0 92.2 95.5 HTL 57.1 68.8 78.7 86.5 92.5 95.5 Table 3 .2 Comparison with the

state-of-art on the CUB-200-2011 dataset [111]. highly improved. The proposed hierarchical loss get 80.9% Recall@1, which is 4.0% higher than BIER [73] and 18.8% higher than HDC [126]. 3 .5.2 Caltech-UCSD Birds 200-2011 Datasets and performance measures. The Caltech-UCSD Birds 200 dataset (CUB-200- 2011) [111] contains photos of 200 bird species with 11788 images. CUB-200-2011 serves as a benchmark in most existing work on deep metric learning and image retrieval. The first 100 classes (5864 images) are used for training, and the rest (5924 images) of classes are used for testing. The rest images are treated as both the query set and the gallery set. For the evaluation, we use the same Recall@K metric as in Section In-Shop Clothes Retrieval. Here, K ∈ {1, 2, 4, 8, 16, 32}. Implementation details. The dimension d of the feature embedding is 512. The triplet violate margin is set to 0.2. As in the previous section, the hierarchical tree is still set to 16 levels. All the training details are almost the same with the In-Shop Clothes Retrieval dataset. But since there are only 100 image classes for training, the dataset is very easy to get overfitting. When we train 10 epoches, the training stopped. The batch size is set to 50. For every 3 epoch, we decrease the learning rate by multiplying 0.1. Result comparison. Table 3.2 lists the results of image retrieval on Caltech-UCSD Birds 200- 2011. The baseline — triplet loss already get the state-of-art results with 55.9% Recall@1 compared with the previous state-of-art HDC 54.6% and BIER 55.3%. If we use the anchor- neighbor sampling and the hierarchical loss, we get 57.1% Recall@1. Since there are only 100 classes and 6000 images for training, the network is very easy to get overfitting. The performance gain gotten by the hierarchical loss is only 1.2% Recall@1. 40 Deep Metric Learning with Hierarchical Triplet Loss 3 .5.3 LFW Face Verification Datasets and performance measures. The CASIA-WebFace dataset [123] is one of the publicly accessible datasets for face recognition. It has been the most popular dataset for the training of face recognition algorithms, such as in [1, 116, 64]. CASIA-WebFace has 10575 identities and 494414 images. We following the testing protocol in [123] to test the performance of our algorithms. The face verification results on LFW dataset [43] is reported. Implementation details. Since the triplet loss is very sensitive to the noise, we clear the CASIA-WebFace using the pre-trained model of VGG-Face [77] and manually remove some noises. About 10% images are removed. Then the remained faces are used to train a SoftMax classifier. The network parameters are initialized by a pre-trained ImageNet model. We fine-tune the pre-trained classification network for face recognition using the hierarchical loss. Result comparison. The triplet loss gets 98.3% accuracy on the LFW face verification task, which is 1.12% lower than the SpereFace[64] — 99.42% which uses the same dataset for training. When we substitute the triplet loss with the hierarchical triplet loss, the results comes to 99.2. It's comparable with state-of-art results. This indicates that the hierarchical triplet loss has stronger discriminative power than triplet loss. While, since the triplet based method are very sensitive to noise, the hierarchical triplet loss get inferior performance compared with SphereFace [64] 99.42% and FaceNet [89] 99.65% . 3 .5. 4 Sampling Matter and Local Optima Sampling Matter. We investigate the influence of batch size on the test set of In-Shop Clothes Retrieval. Fig. 3.5 (a) shows that when the batch size grows from 60 to 480, the accuracy increases in the same iterations. When the training continues, the performance will fluctuates heavily and get overfitting. Besides, when come to the same results at 60% Recall@1, both the anchor-neighbor sampling with triplet loss and the hierarchical loss converge at about 2 times faster than random sampling (Batch Size = 480). Fig. 3.5 (b) shows the compares the convergence speed of the triplet loss (our baseline), the hierarchical triplet loss and the HDC [126] on the test set of Caltech-UCSD Birds 200. Compared to the 60000 iterations (see in [126]), the hierarchical

triplet loss converges in 1000 iterations. The hierarchical triplet loss with anchor-Neighborhood sampling converge faster traditional and get better performance than HDC [126]. Pool Local Optima. In Table 3.1 and Table 3.2, we can find that the triplet loss get inferior performance than the hierarchical triplet loss on both the In-Shop Clothes Retrieval and Caltech-UCSD Birds 200. In the Fig. 3.5, the accuracy of the triplet loss start to fluctuate when the training continues going after the loss drops to very low. In fact, there are always 3.5 Experimental Results and Comparisons 41 (a) Image Retrieval Results on In-Shop Clothes (b) Image Retrieval Results on CUB 200-2011 Fig. 3.5 (a) Image retrieval result comparison on In-Shop Clothes [65] with different batch sizes. (b) Image retrieval result comparison on CUB-200-2011 [111]. very few or zeros triplets in mini-batch even when the network isn't gotten the best results. Then they don't produce gradients and will decay the learnable parameters in networks by SGD [75]. So we incorporate the hierarchical structure to make points in the mini-batch know the position of point that are already far away, and then attempt to push them further from itself and its neighborhood classes. 3.5.5 Ablation Study (a) Triplet Loss (b) Anchor Neighbor Sampling (c) Hierarchical Triplet Loss Fig. 3.6 Visualize part of the In-Shop Clothes Retrieval training set [65]. (a) Triplet Loss. (b) Anchor-Neighbor Sampling. (b) Hierarchical Triplet Loss. 42 Deep Metric Learning with Hierarchical Triplet Loss We perform an ablation study on In-Shop Clothes Retrieval test set by replacing or removing a single component in our pipeline every time. First, to verify the importance of the hierarchical triplet loss, we remove the dynamic margins in the hierarchical triplet loss. Then the margin between any two classes are the same margin α . The R@1 is 75.3% dropped by 5.6% as shown in Table 3.1. Second, the anchor-neighborhood sampling step is removed. We directly sample training classes randomly to form a mini-batch. The R@1 is decreased by 13%. In Fig. 3.6, we can find that the features produced by the anchor-neighborhood sampling with traditional triplet loss has better discriminative ability than triplet loss. When we use the hierarchical triplet loss, the feature points belong to the same class becomes much more compact and don't mixed with others as in the anchor neighbor sampling. 3.6 Conclusions We have presented a new hierarchical triplet loss which is able to select informative training samples (triplets) via an adaptively-updated hierarchical tree that encodes global context. The hierarchical triplet loss effectively handles the main limitation of random sampling, which is a critical issue for deep metric learning. First, we construct a hierarchical tree at the class level which encodes global context information over the whole dataset. Visual similar classes are merged recursively to form the hierarchy. Second, the problem of triplet collection is formulated by proposing a new violate margin. The proposed violate margin is computed dynamically based on the designed hierarchical tree, allowing it to learn from more meaningful hard samples with the guide of global context. The proposed violate margin is easily implemented, can be readily applicable to the standard triplet loss and other deep metric learning approached. The proposed hierarchical triplet loss is evaluated on the tasks of image retrieval and face recognition, where it achieves new state-of-the-art performance on a number of standard benchmarks. Chapter 4 Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning 4 .1 Introduction Deep neural networks give rise to many breakthroughs in computer vision by usinging huge amounts of labeled training data. Supervised object detection and semantic segmentation require object or even pixel level annotations, which are much more labor-intensive to obtain than image level labels. On the other hand, when there exist image level labels only, due to incomplete annotations, it is very challenging to predict accurate object locations, pixel-wise labels, or even image level labels in multi-label image classification.

Given image level supervision only, researchers have proposed many weakly supervised algorithms for detecting objects and labeling pixels. These algorithms employ different mechanisms, including bottom-up, top-down [128, 58] and hybrid approaches [85], to dig out useful information. In bottom-up algorithms, pixels are usually grouped into many object proposals, which are further classified, and the classification results are merged to match groundtruth image labels. In top-down algorithms, images first go through a forward pass of a deep neural network, and the result is then propagated backward to discover which pixels actually contribute to the final result [128, 58]. There are also hybrid algorithms [85] that consider both bottom-up and top-down cues in their pipeline. Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and 44 Semantic Segmentation Based on Weakly Supervised Learning Although there exist many weakly supervised algorithms, the accuracy achieved by top weakly supervised algorithms is still significantly lower than their fully supervised counterparts. This is reflected in both the precision and recall of their results. In terms of precision, results from weakly supervised algorithms contain much more noise and outliers due to indirect and incomplete supervision. Likewise, such algorithms also achieve much lower recall because there is insufficient labeled information for them to learn comprehensive feature representations of target object categories. However, different types of weakly supervised algorithms may return different but complementary subsets of the ground truth. These observations motivate an approach that first collect as many evidences and results as possible from multiple types of solution mechanisms, put them together, and then remove noise and outliers from the fused results using powerful filtering techniques. This is in contrast to deep neural networks trained from end to end. Although this approach needs to collect results from multiple separately trained networks, the filtered and fused evidences are eventually used for training a single network used for the testing stage. Therefore, the running time of the final network during the testing stage is still comparable to that of state-of-the-art end-to-end networks. According to the above observations, we propose a weakly supervised curriculum learning pipeline for object recognition, detection and segmentation. At a high level, we obtain object localization and pixelwise semantic labeling results for the training images first using their image level labels, and then use such intermediate results to train object detection, semantic segmentation, and multi-label image classification networks in a fully supervised manner. Since image level, object level and pixel level analysis has mutual dependencies, they are not performed independently but organized into a single pipeline with four stages. In the first stage, we collect object localization results in the training images from both bottom-up and top-down weakly supervised object detection algorithms. In the second stage, we incorporate both metric learning and density-based clustering to filter detected object instances. In this way, we obtain a relatively clean and complete set of object instances. Given these object instances, we further train a single-label object classifier, which is applied to all object instances to obtain their final class labels. Third, to obtain a relatively clean pixel-wise probability map for every class and every training image, we fuse the image level attention map, object level attention maps and an object detection heat map. The pixel-wise probability maps are used for training a fully convolutional network, which is applied to all training images to obtain their final pixel-wise label maps. Finally, the obtained object instances and pixel-wise label maps for all the training images are used for training deep networks for object detection and semantic segmentation respectively. To make pixel-wise label maps of the training images help multi-label image classification, we perform multi-task 4.2 Related Work 45 learning by training a single deep network with two branches, one for multi-label image

classification and the other for pixel labeling. Experiments show that our weakly supervised curriculum learning system is capable of achieving state-of-the-art results in multi-label image classification as well as weakly supervised object detection and very competitive results in weakly supervised semantic segmentation on MS-COCO [62], PASCAL VOC 2007 and PASCAL VOC 2012 [26]. In summary, this paper has the following contributions. • We introduce a novel weakly supervised pipeline for multi-label object recognition, detec- tion and semantic segmentation. In this pipeline, we first obtain intermediate labeling results for the training images, and then use such results to train task-specific networks in a fully supervised manner. • To localize object instances relatively accurately in the training images, we propose a novel algorithm for filtering, fusing and classifying object instances collected from multiple solution mechanisms. In this algorithm, we incorporate both metric learning and density- based clustering to filter detected object instances. • To obtain a relatively clean pixel-wise probability map for every class and every training image, we propose an algorithm for fusing image level and object level attention maps with an object detection heat map. The fused maps are used for training a fully convolutional network for pixel labeling. 4.2 Related Work Weakly Supervised Object Detection and Segmentation. Weakly supervised object de- tection and segmentation respectively locates and segments objects with image-level labels only [74, 18]. They are important for two reasons: first, learning complex visual concepts from image level labels is one of the key components in image understanding; second, fully supervised deep learning is too data hungry. Methods in [74, 23, 22] treat the weakly supervised localization problem as an image classification problem, and obtain object locations in specific pooling layers of their networks. Methods in [9, 104] extract object instances from images using selective search [108] or edge boxes [133], convert the weakly supervised detection problem into a multi-instance learning problem [19]. The method in [19] at first learns object masks as in [23, 22], and then uses the E-M algorithm to force the network to learn object segmentation masks obtained at previous stages. Since it is very hard for a network to directly learn object locations and pixel Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning (a) Image Level Stage: Proposal Generation (b) Instance Level Stage: Outlier Detection and (c) Pixel Level Stage: Probability Map Fusion and Multi Evidence Fusion Object Instance Filtering and Pixel Label Prediction 46 Input/Image Object Heatmap Image Attention Map Object Instances Triplet Loss Net Filtered Object Instances Label Map with Uncertainty Instance Attention Map Probability Map Instance Classifier Fig. 4 .1 The proposed weakly supervised pipeline. From left to right: (a) Image level stage: fuse the object heatmaps H and the image attention map Ag to generate object instances R for the instance level stage, and provide these two maps for information fusion at the pixel level stage. (b) Instance level stage: perform triplet loss based metric learning and density based clustering for outlier detection, and train a single label instance classifier φs(·,·) for instance filtering. (c) Pixel level stage: integrate the object heatmaps H, instance attention map Al, and image attention map Ag for pixel labeling with uncertainty. 4.3 Weakly Supervised Curriculum Learning 47 labels without sufficient supervision, in this paper, we decompose object detection and pixel labeling into multiple easier problems, and solve them progressively in multiple stages. Neural Attention. Many efforts [128, 6, 58] have been made to explain how neural networks work. The method in [58] extends layer-wise relevance propagation (LRP) [3] to comprehend inherent structured reasoning of deep neural networks. To further ignore the cluttered background, a positive neural attention back-propagation scheme, called excitation back- propagation

(Excitation BP), is introduced in [128]. The method in [6] locates top activations in each convolutional map, and maps these top activation areas into the input image using bilinear interpolation. In our pipeline, we adopt the excitation BP [128] to calculate pixel-wise class probabilities. However for images with multiple category labels, a deep neural network could fuse the activations of different categories in the same neurons. To solve this problem, we train a single-label object instance classification network and perform excitation BP in this network to obtain more accurate pixel level class probabilities. Curriculum Learning. Curriculum learning [7] is part of the broad family of machine learning methods that starts with easier subtasks and gradually increases the difficulty level of the tasks. In [7], Yoshua et al. describe the concept of curriculum learning, and use a toy classification problem to show the advantage of decomposing a complex problem into several easier ones. In fact, the idea behind curriculum learning has been widely used before [7]. Hinton et al. [41] trained a deep neural network layer by layer using a restricted Boltzmann machine [96] to avoid the local minima in deep neural networks. Many machine learning algorithms [100, 32] follow a similar divide-and-conquer strategy in curriculum learning. In this paper, we adopt this strategy to decompose the pixel labeling problem into image level learning, object instance level learning and pixel level learning. All the learning tasks in these three stages are relatively simple using the training data in the current stage and the output from the previous stage. 4.3 Weakly Supervised Curriculum Learning 4.3.1 Overview Given an image I associated with an image level label vector yI = [y1 , y2 , ..., yC]T , our weakly supervised curriculum learning aims to obtain pixel-wise labels Y I = [y1 , y2 , ..., yP]T , and then use these labels to assist weakly supervised object detection, semantic segmentation and multi-label image classification. Here C is the total number of object classes, P is the total number of pixels in I , and yl is binary. yl = 1 means the l -th object class exists in I , and yl = 0 otherwise. The label of a pixel p is denoted by a C -dimensional binary vector 48 Semantic Segmentation Based on Weakly Supervised Learning (a) Heatmap Proposals (b) Attention Proposals (c) Fused Proposals Fig. 4 .2 (a) Proposals Rh and Rl generated from an object heatmap, (b) proposals generated from an attention map, (c) filtered proposals (green), heatmap proposals (red and blue), and attention proposals (purple). yp. The number of object classes existing in I, which is the same as the number of positive components of yI is denoted by K. Following the divide-and-conquer idea in curriculum learning [7], we decompose the pixel labeling task into three stages: the image level stage, the instance level stage and the pixel level stage. 4.3.2 Image Level Stage The image level stage not only decomposes multi-label image classification into a set of single-label object instance classifications, but also provides an initial set of pixel-wise probability maps for the pixel level stage. Object Heatmaps. Unlike the fully supervised case, weakly supervised object detection produces object instances with higher uncertainty and also misses a higher percentage of true objects. To reduce the number of missing detections, we propose to compute an object heatmap H for every object class existing in the image. For an image I with width W and height H, a dense set of object proposals R = (R1,R2,...,Rn) are generated using sliding anchor windows. And the feature stride λs is set to 8. The number of locations in the input image where we can place anchor windows is H/λs ×W/λs. Denote the short side of image I by Lρ. Following the set- ting used for RPN [81], we let the anchor windows at a single location have four scales 4.3 Weakly Supervised Curriculum Learning 49 [Lρ /8, Lρ /4, Lρ /2, Lρ] and three aspect ratios [0.5, 1, 2]. After proposals out of image borders have been removed, there are usually 12000 remaining proposals per image. Here we define a stack of object heatmaps H = [H 1, H 2, ..., H C] as a C × H × W matrix, and all values are set to zero initially. The object

detection and classification network φd (·, ·) used here is the weakly supervised object testing net VGG-16 from [104]. For every proposal Ri in R, its object class probability vector φd (I, Ri) is added to all the pixels in the corresponding window in the heatmaps. Then every heatmap is normalized to [0, 1] as follows, Hc = (Hc − min(Hc))/max(Hc), where H c is the heatmap for the c-th object class. Note that only the heatmaps for object classes existing in I are normalized. All the other heatmaps are ignored and set to zeros. Multiple Evidence Fusion. The object heatmaps highlight the regions that may contain objects even when the level of supervision is very weak. However, since they are generated using sliding anchor windows at multiple scales and aspect ratios, they tend to highlight pixels near but outside true objects, as shown in Fig 4.2. Given an image classification network trained using the image level labels (here we use GoogleNet V1 [128]), neural attention calculates the contribution of every pixel to the final classification result. It tends to focus on the most influential regions but not necessarily the entire objects. Note that false positive regions may occur during excitation BP [128]. To obtain more accurate object instances, we integrate the top-down attention maps Ag = [A1g, A2g, ..., ACg] with the object heatmaps H = [H1, H2, ..., HC]. For object classes existing in image I, their corresponding heatmaps H and attention maps Ag are thresholded by distinct values. The heatmaps H are too smooth to indicate accurate object boundaries, but they provide important spatial priors to constrain object instances obtained from the attention maps. We assume that regions with a sufficiently high value in the object heatmaps should at least include parts of objects, and regions with sufficiently low values everywhere do not contain any objects. Following this assumption, we threshold the heatmaps with two values 0.65 and 0.1 to identify highly confident object proposals Rh = (Rh1, Rh2, ..., RhNh) and relatively low confident object proposals Rl = (Rl1, Rl2, ..., RlNl) after connected component extraction. Then the attention maps are thresholded by 0.5 to attention proposals Ra = (Ra1, Ra2, ..., RaNa) as shown in Fig 4.2. Nh, Nl and Na are the proposal numbers of Rh, Rl and Ra. All these object proposals have corresponding class labels. During the fusion, for each object class, the attention proposals Ra which cover more than 0.5 of any proposals in Rh are preserved. We denote these proposals by R, each of which is modified 50 Semantic Segmentation Based on Weakly Supervised Learning (a) Input Proposals (b) Distance Map Fig. 4 .3 (a) Input proposals of the triplet-loss network, (b) distance map computed using features from the triplet-loss network. slightly to completely enclose the corresponding proposal in Rh meanwhile be completely contained inside the corresponding proposal in Rl (Fig 4.2). 4.3.3 Instance Level Stage Since multiple object categories present in the same image make it hard for neural attention to obtain an accurate pixel-wise attention map for each class, we train a single-label object instance classification network and compute attention maps in this network to obtain more accurate pixel level class probabilities. The fused object instances from the image level stage are further filtered by metric learning and density-based clustering. The remaining labeled object proposals are used for training this object instance classifier, which can also be used to further remove remaining false positive object instances. Metric Learning for Feature Embedding. Metric learning is popular in face recogni- tion [89], person re-identification and object tracking [89, 130, 106]. It embeds an image X into a multi-dimensional feature space by associating this image with a fixed size vector, φt (X , ·), in the feature space. This embedding makes similar images close to each other and dissimilar images apart in the feature space. Thus the similarity between two images can be measured by their distance in this space. The triplet-loss network φt (·, ·) proposed in [89] has the additional property that it can well separate classes even when intra-class distances have large variations. When there exist training samples associated

with incorrect class labels, 4.3 Weakly Supervised Curriculum Learning 51 the loss stays at a high value and the distances between correctly labeled and mislabeled samples remain very large even after the training process has run for a long time. Now let R = [R1,R2,...,RO]T denote the fused object instances from all training images in the image level stage, and Y = [y1,y2,...,yO]T are their labels. Here O is the total number of fused instances, and yl is the label vector of instance Rl. We train a triplet-loss network φt(·,·) using GoogleNet V2 with BatchNorm as in [89]. Each mini-batch first chooses b object classes randomly, and then chooses a instances from these classes randomly. These instances are cropped out from the training images and fed into φt(·,·). Fig. 4.3 visualizes a mini-batch composition and the corresponding pairwise distances among instances. Clustering for Outlier Removal. Clustering aims to remove outliers that are less similar to other object instances in the same class. Specifically, we perform density based clustering [84] to form a single cluster of normal instances within each object class independently, and instances outside this cluster are considered outliers. This is different from that in [84]. Let Rc denote instances in R with class label c, and Nc is the number of instances in Rc. Calculate the pairwise distances d(·,·) among these instances, and obtain the Nc by Nc distance matrix Dc. For an instance Rnc, if its distance from another instance is less than λd (= 0.8), its density dnc is increased by 1. Rank these instances by their densities in a descending order, and choose the instance ranked at the top as the seed of the cluster. Then add instances to the cluster following the descending order if their distance to any element in the cluster is less than λd and their density is higher than Nc/4. Instance Classifier for Re-labeling. Since metric learning and clustering screen object instances in an aggressive way and may heavily decrease their recall, we use the normal instances surviving the previous clustering step to train an instance classifier, which is in turn used to re-label all object proposals generated in the image level stage again. This is a single-label classification problem as each object instance is allowed a single label. GoogleNet V1 with the SoftMax loss serves as the classifier φs(·, ·), and it is fine-tuned from the image level classifier. For every object proposal generated in the previous image level stage, if its label predicted by the instance classifier does not match its original label, it is labeled as an outlier and permanently discarded. 4.3.4 Pixel Level Stage In previous stages, we have already built an image classifier, a weakly supervised object detector, and an object instance classifier. Each of these deep networks produces its own inference result from the input image. For example, the image classifier generates a global attention map, and the object detector generates the object heatmaps. In the pixel level stage, 52 Semantic Segmentation Based on Weakly Supervised Learning (a) Input/Image (b) Object Heatmap (c) Image Attention (d) Instance Attention (e) Probability (e) Segmentation Fig. 4 .4 The pixel labeling process in the pixel level stage. White pixels in the last column indicate pixels with uncertain labels. we still perform multi-evidence filtering and fusion to integrate the inference results from all these component networks to obtain the pixelwise probability map indicating potential object categories at every pixel. The global attention map Ag from the image classifier has a full knowledge about the objects in an image but sometimes only focuses on the most important object parts. The object instance classifier has a local view of each individual object. With the help of object-specific local attention maps generated from the instance classifier, we can avoid missing small objects. Instance Attention Map. Here we define the instance attention map Al as a C ×H ×W matrix, and all values are zero initially. For every surviving object instance from the instance level stage, the object instance classifier φs(·,·) is used to extract its local attention map, and add it to the corresponding region in the instance attention map Al. Normalize the range of Al to [0, 1] as we did for object

heatmaps. Probability Map Integration. The final attention map A is obtained by calculating the element-wise maximum between the image attention map Ag and the instance attention map Al. That is, A = max(Al,Ag). For both the heatmap H and the attention map A, only the classes existing in the image are considered. The background maps of A and H are defined as follows, A0 = max(0,1−ΣCl=1ylAl), H0 = max(0,1−ΣCl=1ylHl). Now both A and H become (C +1)×H ×W matrices. For the l-th channel, if yl = 0, Al = 0 and Hl = 0. Then we perform softmax on both maps along the channel dimension independently. The final probability map P is defined as the result of applying normalization in the class channel to the element-wise product between A and H by treating H as a filter. That is, P = normalize(H ⊙A). Pixel Labeling with Uncertainty. Pixel labels Y I are initialized with the probability map P. For every pixel p, if the maximum element in its label vector y p is larger than a threshold 4.4 Object Recognition, Detection and Segmentation 53 (=0.8), we simply set the maximum element to 1 and other elements to 0; otherwise, the class label at p is uncertain. To inspect these uncertain pixels more carefully, we obtain additional evidence by computing their saliency scores S (normalized into [0, 1]) using an existing state-of-the-art salient object detection algorithm [60]. Given an uncertain pixel q with a high saliency score (Sq ⩾ 0.3), if the maximum element in its label vector yq is larger than a threshold (=0.6) and this element does not correspond to the background, we set the maximum element to 1 and other elements to 0. Given another uncertain pixel o with a low saliency score (So <0.3), if the maximum element in its label vector yo corresponds to the background, we set the background element to 1 and other elements to 0. 4.4 Object Recognition, Detection and Segmentation 4. 4.1 Semantic Segmentation Given pixel-wise labels generated at the end of the pixel level stage for all training images, we train a fully convolutional network (FCN) similar to the network in [66] to perform semantic segmentation. Note that all pixels with uncertain class labels are excluded during training. In the prediction part, we adopt atrous spatial pyramid pooling as in [14]. The resulting trained network can be used for labeling all pixels in any testing image as well as pixels with uncertain labels in all training images. 4. 4.2 Object Detection Once all pixels with uncertain labels in the training images have been re-labeled using the above network for semantic segmentation, we generate object instances in these images by computing bounding boxes of connected pixels sharing the same semantic label. As in [104] and [59], we train fast RCNN [29] using these bounding boxes and their associated labels. Since the bounding boxes generated from the semantic label maps may contain noise, we filter them using our object instance classifier as in Section 4 .3. 3. VGG-16 is still the base network of our object detector, which is trained with five scales and flip as in [104]. 4. 4.3 Multi-label Classification The main component in our multi-label classification network is the structure of ResNet- 101 [38]. There are two branches after layer res4b22_relu of the main component, one branch for classification and the other for semantic segmentation. Both branches share the same structure after layer res4b22_relu. Here we adopt multi-task learning to train both branches. The idea is using the training data for the segmentation branch to make the convolutional 54 Semantic Segmentation Based on Weakly Supervised Learning kernels in the main component more discriminative and powerful. This network architecture is shown in the supplemental materials. Layer pool5 of ResNet-101 in the classification branch is removed, and the output X (∈ R14×14×2048) of layer res5c is a 14 × 14 × 2048 matrix. X is directly fed into a 2048 × 1 × 1 × C convolutional layer, and a classification map $\hat{Y}$ cls(∈ R14×14×C) is obtained. We let the semantic label map $\hat{Y}$ seg(∈ R14×14×C) play the role of an attention map $\hat{Y}$ att after the summation over each channel of the semantic label map is normalized to 1. The final image level probability vector $\hat{y}$ is the

result of spatial average pooling over the element-wise product between Ŷ cls and Ŷ att . Here Ŷ att is used to identify important image regions and assign them larger weights. At the end, the probability vector ŷ is fully connected to an output layer, which performs binary classification for each of the C classes. The cross-entropy loss is used for training the multi-label classification network. The segmentation branch uses atrous spatial pyramid pooling to perform semantic segmentation, and softmax is applied to enforce a single label per pixel. 4.5 Experimental Results Fig. 4.5 The detection and semantic segmentation results on Pascal VOC 2012 test set (the first row) and Pascal VOC 2007 test set (the second row). The detection results are gotten by select proposals with the highest confidence of every class. The semantic segmentation results are post-processed by CRF [54]. All our experiments are implemented using Caffe [47] and run on an NVIDIA TITAN X(Maxwell) GPU with 12GB memory. The hyper-parameters in Section 3 are set according to common sense and confirmed after we visually verify that the segmentation results on a few training samples are valid. The same parameter setting is used for all datasets and has not been tuned on any validation sets. 4.5 Experimental Results 55 4 .5.1 Semantic Segmentation Datasets and performance measures. The Pascal VOC 2012 dataset [25] serves as a benchmark in most existing work on weakly-supervised semantic segmentation. It has 21 classes and 10582 training images (the VOC 2012 training set and additional data annotated in [35]), 1449 for validation and 1456 for testing. Only image tags are used as training data in our experiments. We report results on both the validation (supplemental materials) and test sets. Implementation details. Our network is based on VGG-16. The layers after relu5_3 and layer pool4 are removed. Dilations in layers conv5_1, conv5_2, and conv5_3 are set to 2. The feature stride λs at layer relu5_3 is 8. We add the atrous spatial pyramid pooling as in DeepLab V3 [14] after layer relu5_3. The dilations in our atrous spatial pyramid pooling layers are [1,2,4,6]. This FCN is implemented in py-faster-rcnn [82]. For data augmentation, we use five image scales (480, 576, 688, 864, 1024) (the shorter side is resized to one of these scales) and horizontal flip, and cap the longer side at 1200. During testing, the original size of an input image is preserved. The network is fine-tuned from the pre-trained model for ImageNet in [95]. The learning rate γ is set to 0.001 in the first 20k iterations, and 0.0001 in the next 20k iterations. The weight decay is 0.0005, and the mini-batch size is 1. Post-processing using CRF [54] is added during testing. Result comparison. We compare our method with existing state-of-the-art algorithms. Table 4.1 lists the results of weakly supervised semantic segmentation on Pascal VOC 2012. The proposed method achieves 55.6% mean IoU, comparable to the state of the art (AE- SPL [?]). Recent algorithms, including AE-PSL[?], F-B [88], FCL [85], and SEC [53], all conduct end-to-end training to learn object score maps. Our method demonstrates that if we filter and integrate multiple types of intermediate evidences at different granularities during weakly supervised training, the results become equally competitive or even better. 4.5.2 Object Detection Datasets and performance measures. The performance of our object detector in Sec- tion 4. 4.2 is evaluated on the popular Pascal VOC 2007 and Pascal VOC 2012 datasets [25]. Each of these two datasets is divided into train, val and test sets. The trainval sets (5011 images for 2007 and 11540 images for 2012) are used for training, and only image tags are used. Two measures are used to test our model: mAP and CorLoc. According to the standard Pascal VOC protocol, the mean average precision (mAP) is used for testing our trained models on the test sets, and the correct localization (CorLoc) is used for measuring the object localization accuracy [17] on the trainval sets whose image tags are already used as training data. 56 Semantic Segmentation Based on Weakly Supervised Learning Implementation details. We use the

code for py-faster-rcnn [82] to implement fast R- CNN [29]. The network is still VGG-16. The learning rate is set to 0.001 in the first 30k iterations, and 0.0001 in the next 10k iterations. The momentum and weight decay are set to 0.9 and 0.0005 respectively. We follow the same data augmentation setting in [104], use five image scales (480, 576, 688, 864, 1200) and horizontal flip, and cap the longer image side at 2000. Result comparison. Object detection results on Pascal VOC 2007 test set (Table 4.5) and Pascal VOC 2012 test set (supplemental materials) are reported. Object localization results on Pascal VOC 2007 trainval set and Pascal VOC 2012 trainval set are also reported (supplemental material). On Pascal VOC 2012 test set, our algorithm achieves the highest mAP (47.5%), at least 5.0% higher than the latest state-of-the-art algorithms including OICR [104] and HCP+DSD+OSSH3[48]. Our trained model also achieves the highest mAP (51.2%) among all weakly supervised algorithms on Pascal VOC 2007 test set, 4.2% higher than the latest result from [104]. The object localization accuracy (CorLoc) of our trained model on Pascal VOC 2007 trainval set and Pascal VOC 2012 trainval set are respectively 67% and 69.4%, which are 2.7% and 3.8% higher than the previous best. 4.5.3 Multi-Label Classification Dataset and performance measures. Microsoft COCO [62] is the most popular dataset in multi-label classification. MS-COCO was primarily built for object recognition tasks in the context of scene understanding. The training set is composed of 82081 images in 80 classes, on average 2.9 object labels per image. Since the groundtruth labels of the test set is not available, performance evaluation is conducted on the validation set with 40504 images. We train our models on the training set and test them on the validation set. Performance measures for multi-label classification is quite different from those for single- label classification. Following [132, 115], we employ macro/micro precision, macro/micro recall, and macro/micro F1-measure to evaluate our trained models. For precision, recall and F1-measure, labels with confidence higher than 0.5 are considered positive. "P-C", "R-C" and "F1-C" represent the average per-class precision, recall and F1-measure while "P-O", "R-O" and "F1-O" represent the average overall precision, recall and F1-measure. These measures do not require a fixed number of labels per image. To compare with existing state-of-the-art algorithms, we also report the results of top-3 labels with confidence higher than 0.5 as in [115]. Implementation details. Our main network for multi-label classification is ResNet-101 as described earlier. The resolution of the input images is at 448 × 448. We first train a network with the classification branch only. As a common practice, a pre-trained model for ImageNet 4.6 Multi-label Image Classification 57 is fine-tuned with the learning rate γ set to 0.001 in the first 20k iterations, and 0.0001 in the next 20k iterations. The weight decay is 0.0005. Then we add the segmentation branch and train this new branch only by fixing all the layers before layer res4b22_relu and the classification branch. The learning rate is set to 0.001 in the frist 20k iterations, and 0.0001 in the next 20k iterations. At last, we train the entire network with both branches using the cross-entropy loss for multi-label classification for 30k iterations with a learning rate 0.0001 while still fixing the layers before layer res4b22_relu. Result comparison. In addition to our two-branch network, we also train a ResNet-101 classification network as our baseline. The multi-label classification performance of both networks on MS-COCO is reported in Table 4.3. Since the input resolution of our baseline is 448×448, in comparison to the latest work (ResNet101-SRN) [132], the performance of our baseline is slightly better. Specifically, the F1-C of our baseline is 72.8%, which is 2.8% higher than the F1-C of ResNet101-SRN. In comparison to the baseline, our two-branch network further achieves overall better performance. Specifically, the P-C of our two-branch network is 6.6% higher than the baseline, the R-C is 2.7% lower, and the F1-C is 2.1% higher.

All F1-measures (F1-C, F1-O, F1-C/top3 and F1-O/top3) of our two-branch network are the highest among all state-of-the-art algorithms. 4.6 Multi-label Image Classification Fig 4.6 shows the pipeline of our multi-label classification network. Layers before res4b22_relu of ResNet-101 are shared by the following branches. Both of the segmentation branch and the attention branch have the same structure of the res5 part of ResNet-101. In the classification branch, the output X (∈ R14×14×2048) of layer res5c is a 14 × 14 × 2048 matrix. Then the clas- sification map $\hat{Y}$ cls(∈ R14 ×14×C) is obtained by feeding X directly into a 2048 × 1 × 1 × C convolutional layer. In the segmentation branch, the output of layer res5c is fed into an atrous spatial pyramid pooling layer, and then a 1280 × 1 × 1 × C convolutional layer and a softmax layer to get the segmentation map $\hat{Y}$ seg(∈ R14 ×14×C). Normalize the summation of each channel in $\hat{Y}$ seg to get the attention map $\hat{Y}$ att . In our atrous spatial pyramid pooling layer, we have four dilated convolution layers and one global convolution layer. The dilation of the four dilated convolution layer are [1, 2, 4, 6]. All these convolution layers have 256 channels. 4.6.1 Ablation Study We perform an ablation study on Pascal VOC 2007 detection test set by replacing or removing a single component in our pipeline every time. First, to verify the importance of object 58 Semantic Segmentation Based on Weakly Supervised Learning Fig. 4.6 The architecture of the multi-label classification. … 4.7 Conclusions 59 instances, we remove all steps related to object instances, including the entire instance level stage and the operations related to the instance attention map in the pixel level stage. The mAP is decreased by 3.1% as shown in Table 4.5. Second, the clustering and outlier detection step in the instance level stage is removed. We directly train an instance classifier using the object proposals from the image level stage. The mAP is decreased by 2.7%. Third, instead of labeling a subset of pixels only in the pixel level stage, we assign a unique label to every pixel even in the case of low confidence. The mAP drops to 47.5%, 3.7% lower than the performance of the original pipeline. 4.7 Conclusions In this paper, we have presented a new pipeline for weakly supervised object recognition, detection and segmentation. Different from previous algorithms, we fuse and filter object instances from different techniques and perform pixel labeling with uncertainty. We use the resulting pixel-wise labels to generate groundtruth bounding boxes for object detection and attention maps for multi-label classification. Our pipeline has achieved clearly better performance in all of these tasks. Nevertheless, how to simplify the steps in our pipeline deserves further investigation. Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning method bg aero bike bird boat bottle bus car cat chair cow table dog horse mbike person plant sheep sofa train tv mIoU 60 SEC[53] 83.5 56.4 28.5 64.1 23.6 46.5 70.6 58.5 71.3 23.2 54.0 28.0 68.1 62.1 FCL[85] 85.7 58.8 30.5 67.6 24.7 44.7 74.8 61.8 73.7 22.9 57.4 27.5 71.3 64.8 TP-BM[51] 83.4 62.2 26.4 71.8 18.2 49.5 66.5 63.8 73.4 19.0 56.6 35.7 69.3 61.3 AE-PSL[?] - - - - - - - - - - - - - - - - - 70.0 55.0 72.4 57.3 71.7 69.2 - - 38.4 58.0 39.9 38.4 48.3 51.7 37.0 60.4 42.8 42.2 50.6 53.7 39.1 66.3 44.8 35.9 45.5 53.8 - - - - - 55.7 Ours+CRF 86.6 72.0 30.6 68.0 44.8 46.2 73.4 56.6 73.0 18.9 63.3 32.0 70.1 72.2 68.2 56.1 34.5 67.5 29.6 60.2 43.6 55.6 Table 4.1 Comparison among weakly supervised semantic segmentation methods on PASCAL VOC 2012 segment at ion t est set. 4.7 Conclusions method aero bike bird boat bottle bus car cat chair cow table dog horse mbike person plant sheep sofa train tv mAP OM+MIL+FRCNN[59] HCP+DSD+OSSH3[48] OICR-Ens+FRCNN[104] 54.5 47.4 41.3 20.8 17.7 51.9 63.5 46.1 21.8 57.1 22.1 34.4 50.5 54.2 52.0 35.2 25.9 15.0 59.6 67.9 58.7 10.1 67.4 27.3 37.8 54.8 65.5 67.2 47.2 21.6 22.1 68.0 68.5 35.9 5.7 63.1 49.5 30.3 64.7 61.8 67.3 66.1 16.2 5.1 13.0 29.9 40.7 15.9 55.3 40.2 39.5 19.7 52.6 43.5 56.9 62.5 43.7 25.6 50.0 57.1

60.2 59.0 47.0 Ours+FRCNN w/o clustering 66.7 61.8 55.3 41.8 6.7 61.2 62.5 72.8 12.7 46.2 40.9 71.0 67.3 Ours+FRCNN w/o uncertainty 66.8 63.4 54.5 42.2 5.8 60.5 58.3 67.8 7.8 46.1 40.3 71.0 68.2 Ours+FRCNN w/o instances 67.7 62.9 53.1 44.4 11.2 62.4 58.5 71.2 8.3 45.7 41.5 71.0 68.0 Ours+FRCNN 64.3 68.0 56.2 36.4 23.1 68.5 67.2 64.9 7.1 54.1 47.0 57.0 69.3 64.7 30.9 62.6 30.7 59.2 30.3 65.4 20.8 16.7 42.6 56.0 65.0 26.5 48.5 16.5 41.1 55.2 66.8 25.2 47.5 15.0 42.4 56.0 67.2 26.8 48.1 23.2 50.7 59.6 65.2 57.0 51.2 Table 4 .2 Average precision (in %) of weakly supervised methods on PASCAL VOC 2007 detection test set. 61 62 Semantic Segmentation Based on Weakly Supervised Learning method F1-C P-C R-C F1-O P-O R-O F1-C/top3 P-C/top3 R-C/top3 F1-O/top3 P-O/top3 R-O/top3 CNN-RNN[114] RLSD[129] RNN-Attention[115] ResNet101-SRN[132] - - - - - - 70.0 81.2 - - - - - - 63.3 75.0 - - - - - - 84.1 67.7 60.4 66.0 62.0 67.6 67.4 79.1 66.3 85.8 55.6 67.8 57.2 66.5 58.7 72.0 57.5 72.1 69.2 70.1 84.0 88.1 66.4 63.4 63.0 61.1 ResNet101(448 × 448)(baseline) 72.8 73.8 72.9 76.3 77.5 75.1 69.5 78.3 63.7 73.1 83.8 64.9 Ours 74.9 80.4 70.2 78.4 85.2 72.5 70.6 84.5 62.2 74.7 89.1 64.3 Table 4.3 Performance comparison among multi-label classification methods on Microsoft COCO 2014 validation set. 4.7 Conclusions method bg aero bike bird boat bottle bus car cat chair cow table dog horse mbike person plant sheep sofa train tv mIoU DSCM[92] 76.7 45.1 24.6 40.8 23.0 34.8 61.0 51.9 52.4 15.5 45.9 32.7 54.9 48.6 F-B[88] 79.2 60.1 20.4 50.7 41.2 46.3 62.6 49.2 62.3 13.3 49.7 38.1 58.4 49.0 SEC[53] 82.4 62.9 26.4 61.6 27.6 38.1 66.6 62.7 75.2 22.1 53.5 28.3 65.8 57.8 FCL[85] 85.8 65.2 29.4 63.8 31.2 37.2 69.6 64.3 76.2 21.4 56.3 29.8 68.2 60.6 T -P[51] 82.8 62.2 23.1 65.8 21.1 43.1 71.1 66.2 76.1 21.3 59.6 35.1 70.2 58.8 57.4 51.8 57.0 48.2 62.3 52.5 66.2 55.8 62.3 66.1 38.2 55.4 32.2 42.6 39.6 44.1 27.8 55.1 29.6 54.6 26.6 46.6 32.2 62.6 32.1 45.4 45.3 50.7 30.8 66.1 34.9 48.8 47.1 52.8 35.8 69.9 33.4 45.9 45.6 53.1 Ours+CRF 85.8 72.5 29.1 66.0 55.7 49.6 73.1 61.4 77.5 26.6 68.5 31.8 73.6 71.5 68.8 53.1 31.8 79.8 35.7 64.9 41.3 58.0 Table 4.4 Comparison among weakly supervised semantic segmentation methods on PASCAL VOC 2012 segmentation val set. 63 Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning method aero bike bird boat bottle bus car cat chair cow table dog horse mbike person plant sheep sofa train tv mAP 64 OM+MIL+FRCNN[59] HCP+DSD+OSSH3[48] OICR-Ens+FRCNN[104] 54.5 47.4 41.3 20.8 17.7 51.9 63.5 46.1 21.8 57.1 22.1 34.4 50.5 54.2 52.0 35.2 25.9 15.0 59.6 67.9 58.7 10.1 67.4 27.3 37.8 54.8 65.5 67.2 47.2 21.6 22.1 68.0 68.5 35.9 5.7 63.1 49.5 30.3 64.7 61.8 16.2 67.3 5.1 66.1 13.0 29.9 40.7 15.9 55.3 40.2 39.5 19.7 52.6 43.5 56.9 62.5 43.7 25.6 50.0 57.1 60.2 59.0 47.0 Ours+FRCNN w/o clustering 66.7 61.8 55.3 41.8 6.7 61.2 62.5 72.8 12.7 46.2 40.9 71.0 67.3 Ours+FRCNN w/o uncertainty 66.8 63.4 54.5 42.2 5.8 60.5 58.3 67.8 7.8 46.1 40.3 71.0 68.2 Ours+FRCNN w/o instances 67.7 62.9 53.1 44.4 11.2 62.4 58.5 71.2 8.3 45.7 41.5 71.0 68.0 Ours+FRCNN w/o filtering 69.0 67.1 53.8 39.3 13.1 61.4 64.3 72.5 15.3 48.0 42.4 67.2 68.0 Ours+FRCNN w/o heatmap 65.9 65.9 57.6 40.3 7.6 61.7 62.7 73.4 11.9 49.2 44.3 68.6 70.8 Ours+FRCNN 64.3 68.0 56.2 36.4 23.1 68.5 67.2 64.9 7.1 54.1 47.0 57.0 69.3 64.7 30.9 62.6 30.7 59.2 30.3 65.5 32.4 64.0 33.6 65.4 20.8 16.7 42.6 56.0 65.0 26.5 48.5 16.5 41.1 55.2 66.8 25.2 47.5 15.0 42.4 56.0 67.2 26.8 48.1 17.1 42.2 55.6 67.0 23.8 49.3 15.2 42.3 54.5 66.1 23.4 49.0 23.2 50.7 59.6 65.2 57.0 51.2 Table 4.5 Average precision (in %) of weakly supervised methods on PASCAL VOC 2007 detection test set. 4.7 Conclusions method aero bike bird boat bottle bus car cat chair cow table dog horse mbike person plant sheep sofa train tv mAP OICR-VGG16[104] 67.7 61.2 41.5 25.6 22.2 54.6 49.7 25.4 19.9 47.0 18.1 26.0 38.9 WSDDN+context[18] 64.0 54.9 36.4 8.1 12.6 53.1 40.5 28.4 6.6 35.3 34.4 49.1 42.6 HCP+DSD+OSSH3+NR[48] 60.8 54.2 34.1 14.9 13.1

54.3 53.4 58.6 3.7 53.1 8.3 43.4 49.8 OICR-Ens+FRCNN[104] 71.4 69.4 55.1 29.8 28.1 55.0 57.9 24.4 17.2 59.1 21.8 26.6 57.8 67.7 2.0 62.4 19.8 69.2 4.1 71.3 1.0 22.6 41.1 34.3 37.9 55.3 37.9 15.2 27.0 33.1 33.0 50.0 35.3 17.5 43.8 25.6 55.0 50.1 38.3 23.1 52.7 37.5 33.5 56.6 42.5 Ours+FRCNN 71.0 66.9 55.9 33.8 24.0 57.6 58.0 61.4 22.5 58.4 19.2 58.7 61.9 75.0 11.2 23.9 50.3 44.9 41.3 54.3 47.5 Table 4.6 Average precision (in %) of weakly supervised methods on PASCAL VOC 2012 detection test set. 65 Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning method aero bike bird boat bottle bus car cat chair cow table dog horse mbike person plant sheep sofa train tv mCorLoc 66 OICR -VGG16[104] 81.7 80.4 48.7 49.5 32.8 81.7 85.4 40.1 40.6 79.5 35.7 33.7 60.5 WSDDN-Ens[18] 68.9 68.7 65.2 42.5 40.6 72.6 75.2 53.7 29.7 68.1 33.5 45.6 65.9 OM+MIL+FRCNN[59] 78.2 67.1 61.8 38.1 36.1 61.8 78.8 55.2 28.5 68.8 18.5 49.2 64.1 HCP+DSD+OSSH3[48] 72. 2 55.3 53.0 27.8 35.2 68.6 81.9 60.7 11.6 71.6 29.7 54.3 64.3 OICR-Ens+FRCNN[104] 85.8 82.7 62.8 45.2 43.5 84.8 87.0 46.8 15.7 82.2 51.0 45.6 83.7 88.8 21.8 86.1 27.5 73.5 21.4 88.2 22.2 91.2 22.2 57.9 76.3 59.9 75.3 81.4 44.9 76.0 62.4 66.3 66.8 47.4 64.6 22.3 60.9 52.3 53.7 72.2 52.6 68.9 74.4 59.7 75.3 65.1 76.8 78.1 60.6 58.0 52.4 54.9 64.3 Ours+FRCNN 88.3 77.6 74.8 63.3 37.8 78.2 83.6 72.7 19.4 79.5 46.4 78.1 84.7 90.4 28.6 43.6 76.3 68.3 77.9 70.6 67.0 Table 4.7 CorLoc (in %) of weakly supervised methods on PASCAL VOC 2007 detection trainval set. 4.7 Conclusions method aero bike bird boat bottle bus car cat chair cow table dog horse mbike person plant sheep sofa train tv mCorLoc OICR -VGG16[104] 86.2 84.2 68.7 55.4 46.5 82.8 74.9 32.2 46.7 82.8 42.9 41.0 68.1 89.6 WSDDN+context[18] 78.3 70.8 52.5 34.7 36.6 80.0 58.7 38.6 27.7 71.2 32.3 48.7 76.2 77.4 HCP+DSD+OSSH3+NR[48] 82.4 68.1 54.5 38.9 35.9 84.7 73.1 64.8 17.1 78.3 22.5 57.0 70.8 86.6 OICR-Ens+FRCNN[104] 89.3 86.3 75.2 57.9 53.5 84.0 79.5 35.2 47.2 87.4 43.4 43.8 77.0 91.0 9.2 53.9 81.0 52.9 59.5 83.2 16.0 48.4 69.9 47.5 66.9 62.9 18.7 49.7 80.7 45.3 70.1 77.3 10.4 60.7 86.8 55.7 62.0 84.7 62.1 54.8 58.8 65.6 Ours+FRCNN 88.0 81.6 75.8 60.9 46.2 85.3 75.3 76.5 47.2 85.4 47.7 74.3 87.8 91.4 21.6 55.3 77.9 68.8 64.9 75.0 69.4 Table 4.8 CorLoc (in %) of weakly supervised methods on PASCAL VOC 2012 detection trainval set. 67 Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and 68 Semantic Segmentation Based on Weakly Supervised Learning method mAP@.5 mAP@[.5, 0.95] OICR-Ens+FRCNN[104](impl. in this paper) 17.4 7.7 Ours+FRCNN 19.3 8.9 Table 4.9 Average precision (in %) of weakly supervised methods on Microsfot COCO 2014 detection validation set. 4.7 Conclusions 69 Fig. 4.7 The detection and semantic segmentation results on Pascal VOC 2007 t est set . The detection results are gotten by select proposals with the highest confidence of every class. The semantic segmentation results are post-processed by CRF [54]. Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and 70 Semantic Segmentation Based on Weakly Supervised Learning Fig. 4.8 The detection and semantic segmentation results on Pascal VOC 2007 t est set . The detection results are gotten by select proposals with the highest confidence of every class. The semantic segmentation results are post-processed by CRF [54]. 4.7 Conclusions 71 Fig. 4.9 The detection and semantic segmentation results on Pascal VOC 2012 t est set . The detection results are gotten by select proposals with the highest confidence of every class. The semantic segmentation results are post-processed by CRF [54]. Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and 72 Semantic Segmentation Based on Weakly Supervised Learning Fig. 4.10 The detection and semantic segmentation results on Pascal VOC 2012 t est set . The detection results are gotten by select proposals with the highest confidence of every class. The semantic segmentation results are

post-processed by CRF [54]. Chapter 5 Conclusion and Future Research In this thesis, we discuss three important issues on learning from imperfect data: classifying images with insufficient training data, learning feature embeddings for image distance calculation, and recognizing pixels from image level annotation. 1) For classifying images with insufficient training data, we address the overfitting prob- lem by introducing a new transfer learning pipeline called selective joint-tuning. It performs a target learning task with insufficient training data simultaneously with another source learning task with abundant training data. By jointly trained with the nearest neighbors retrieved from the source dataset in low feature space, the general- ization ability of deep networks on the target task have been improved. However, how the source learning task help to learn the target task better remains uncertain. It will need more investigation in the parameter learning process during the joint fine-tuning. Besides, the choice of the most suitable source domain for a specific target learning task remains an open problem. 2) For learning feature embeddings for image distance calculation, we aim to use the hierarchical tree to represent the data distribution of a given dataset. Then it's used as a guidance to select informative training samples (triplets) and encode the global context at the same time. Hierarchical triplet loss (HTL) addresses the main limitation of random sampling, which is a critical issue for deep metric learning. Experiments demonstrate such strategy works quite well on various image retrieval and face recog- nition datasets. However, HTL, including other deep metric learning algorithms, emphasize samples from the same class should be closer than that from different classes, and don't take the multi modal distribution of the data in each class into con- sideration. Then how to design algorithms that can catch the multi modal distribution 74 Conclusion and Future Research during deep metric learning need remain to be an open problem, and need to be solved in future works. 3) For recognizing pixels from image level annotation, we divide the weakly supervised problem into three stages — image level stage, object level stage, and pixel level stage, following the divide-and-conquer strategy in curriculum learning. We fuse and filter object instances from different techniques and perform pixel labeling with uncertainty. Then the resulting pixel labels are used to generate groundtruth bounding boxes for object detection, and attention maps for multi-label classification. Experimental results show that we get new state-of-art results on multi-label classification, weakly supervised object detection and semantic segmentation. However, during the estimation of the pixel level labels, there are too many networks to be trained and makes the whole pipeline very complex. In the future works, we will try to simply the whole pipeline, and design an end-to-end learning system to learn the pixel labels efficiently. References [1] Amos, B., Ludwiczuk, B., and Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. CMU School of Computer Science. [2] Azizpour, H., Sharif Razavian, A., Sullivan, J., Maki, A., and Carlsson, S. (2015). From generic to specific deep representations for visual recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 36–45. [3] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7):e0130140. [4] Bai, S., Bai, X., Tian, Q., and Latecki, L. J. (2017a). Regularized diffusion process for visual retrieval. In AAAI, pages 3967–3973. [5] Bai, S., Zhou, Z., Wang, J., Bai, X., Latecki, L. J., and Tian, Q. (2017b). Ensemble diffusion for retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 774 –783. [6] Bau, D., Zhou, B., Khosla, A., Oliva, A., and Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In The IEEE Conference on Computer Vision and Pattern

Recognition (CVPR). [7] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In Proceedings of the 26th annual international conference on machine learning, pages 41–48. ACM. [8] Bickel, S., Brückner, M., and Scheffer, T. (2009). Discriminative learning under covariate shift. Journal of Machine Learning Research, 10(Sep):2137–2155. [9] Bilen, H. and Vedaldi, A. (2016). Weakly supervised deep detection networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2846–2854. [10] Bo, L., Ren, X., and Fox, D. (2013). Multipath sparse coding using hierarchical matching pursuit. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 660–667. [11] Bucher, M., Herbin, S., and Jurie, F. (2016). Hard negative mining for metric learning based zero-shot classification. In European Conference on Computer Vision, pages 524–531. Springer. 76 References [12] Caruana, R. (1998). Multitask learning. In Learning to learn, pages 95–133. Springer. [13] Chapelle, O., Shivaswamy, P., Vadrevu, S., Weinberger, K., Zhang, Y., and Tseng, B. (2011). Boosted multi-task learning. Machine learning, 85(1- 2):149–173. [14] Chen, L.-C., Papandreou, G., Schroff, F., and Adam, H. (2017a). Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587. [15] Chen, W., Chen, X., Zhang, J., and Huang, K. (2017b). Beyond triplet loss: A deep quadruplet network for person re-identification. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [16] Cimpoi, M., Maji, S., Kokkinos, I., and Vedaldi, A. (2016). Deep filter banks for texture recognition, description, and segmentation. International Journal of Computer Vision, 118(1):65–94. [17] Deselaers, T., Alexe, B., and Ferrari, V. (2012). Weakly supervised localization and learning with generic knowledge. International journal of computer vision, 100(3):275– 293. [18] Diba, A., Sharma, V., Pazandeh, A., Pirsiavash, H., and Van Gool, L. (2016). Weakly supervised cascaded convolutional networks. arXiv preprint arXiv:1611.08258. [19] Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. Artificial intelligence, 89(1):31–71. [20] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In ICML, pages 647–655. [21] Duan, L., Xu, D., Tsang, I. W.-H., and Luo, J. (2012). Visual event recognition in videos by learning from web data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(9):1667–1680. [22] Durand, T., Mordan, T., Thome, N., and Cord, M. (2017). Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017). [23] Durand, T., Thome, N., and Cord, M. (2016). Weldon: Weakly supervised learning of deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4743–4752. [24] Eigen, D. and Fergus, R. (2015). Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In Proceedings of the IEEE International Conference on Computer Vision, pages 2650–2658. [25] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1):98–136. [26] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338. References 77 [27] Evgeniou, A. and Pontil, M. (2007). Multi-task feature learning. Advances in neural information processing systems, 19:41. [28] Gavves, E., Fernando, B., Snoek, C. G., Smeulders, A. W., and Tuytelaars, T. (2015). Local alignments for fine-grained categorization. International Journal of Computer Vision, 111(2):191

–212. [29] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448. [30] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587. [31] Gong, B., Grauman, K., and Sha, F. (2013). Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In ICML (1), pages 222–230. [32] Graves, A., Bellemare, M. G., Menick, J., Munos, R., and Kavukcuoglu, K. (2017). Automated curriculum learning for neural networks. arXiv preprint arXiv:1704.03003. [33] Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. [34] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In Computer vision and pattern recognition, 2006 IEEE computer society conference on, volume 2, pages 1735–1742. IEEE. [35] Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., and Malik, J. (2011). Semantic contours from inverse detectors. In Computer Vision (ICCV), 2011 IEEE International Conference on, pages 991–998. IEEE. [36] Harwood, B., Kumar B G, V., Carneiro, G., Reid, I., and Drummond, T. (2017). Smart mining for deep metric learning. In The IEEE International Conference on Computer Vision (ICCV). [37] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385. [38] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778. [39] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. arXiv preprint arXiv:1603.05027. [40] Herranz, L., Jiang, S., and Li, X. (2016). Scene recognition with cnns: objects, scales and dataset bias. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 571–579. [41] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. Neural computation, 18(7):1527–1554. 78 References [42] Hong, S., Oh, J., Han, B., and Lee, H. (2016). Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In IEEE Conference on Computer Vision and Pattern Recognition. [43] 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 07-49, University of Massachusetts, Amherst. [44] Huang, J., Gretton, A., Borgwardt, K. M., Schölkopf, B., and Smola, A. J. (2006). Correcting sample selection bias by unlabeled data. In Advances in neural information processing systems, pages 601–608. [45] Huh, M., Agrawal, P., and Efros, A. A. (2016). What makes imagenet good for transfer learning? arXiv preprint arXiv:1608.08614. [46] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning, pages 448–456. [47] 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 Proceedings of the 22nd ACM international conference on Multimedia, pages 675–678. ACM. [48] Jie, Z., Wei, Y., Jin, X., Feng, J., and Liu, W. (2017). Deep self-taught learning for weakly supervised object localization. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [49] Karen, S. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv :1409.1556. [50] Khosla, A., Jayadevaprakash, N., Yao, B., and Li, F.-F. (2011). Novel dataset for fine-grained image categorization: Stanford dogs. In Proc. CVPR Workshop on Fine-Grained Visual Categorization (FGVC), volume 2. [51] Kim, D., Cho, D., Yoo, D., and So Kweon, I. (2017). Two-phase learning for weakly supervised

object localization. In The IEEE International Conference on Computer Vision (ICCV). [52] Kim, Y.-D., Jang, T., Han, B., and Choi, S. (2015). Learning to select pre-trained deep representations with bayesian evidence framework. arXiv preprint arXiv:1506.02565. [53] Kolesnikov, A. and Lampert, C. H. (2016). Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In European Conference on Computer Vision, pages 695–711. Springer. [54] Krähenbühl, P. and Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In Advances in neural information processing systems, pages 109–117. References 79 [55] Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J., and Fei-Fei, L. (2015). The unreasonable effectiveness of noisy data for fine-grained recognition. arXiv preprint arXiv:1511.06789. [56] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. [57] Kumar, B., Carneiro, G., Reid, I., et al. (2016). Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5385–5394. [58] Lapuschkin, S., Binder, A., Montavon, G., Muller, K.-R., and Samek, W. (2016). Analyzing classifiers: Fisher vectors and deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2912–2920. [59] Li, D., Huang, J.-B., Li, Y., Wang, S., and Yang, M.-H. (2016). Weakly supervised object localization with progressive domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3512–3520. [60] Li, G. and Yu, Y. (2016). Deep contrast learning for salient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 478–487. [61] Li, Z. and Hoiem, D. (2016). Learning without forgetting. In European Conference on Computer Vision, pages 614–629. Springer. [62] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer. [63] Lin, T.-Y., RoyChowdhury, A., and Maji, S. (2015). Bilinear cnn models for fine- grained visual recognition. In Proceedings of the IEEE International Conference on Computer Vision, pages 1449–1457. [64] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., and Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1. [65] Liu, Z., Luo, P., Qiu, S., Wang, X., and Tang, X. (2016). Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1096–1104. [66] 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, pages 3431–3440. [67] Long, M. and Wang, J. (2015). Learning transferable features with deep adaptation networks. CoRR, abs/1502.02791, 1:2. [68] Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605. 80 References [69] Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In 2015 IEEE conference on computer vision and pattern recognition (CVPR), pages 5188–5196. IEEE. [70] Manjunath, B. S. and Ma, W.-Y. (1996). Texture features for browsing and retrieval of image data. IEEE Transactions on pattern analysis and machine intelligence, 18(8):837– 842. [71] Nilsback, M.- E. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on, pages 722–729. IEEE. [72] Opitz, M., Waltner, G., Possegger, H., and Bischof, H.

(2017a). Bier - boosting independent embeddings robustly. In The IEEE International Conference on Computer Vision (ICCV). [73] Opitz, M., Waltner, G., Possegger, H., and Bischof, H. (2017b). Bier-boosting indepen- dent embeddings robustly. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5189–5198. [74] Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2015). Is object localization for free?- weakly-supervised learning with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 685–694. [75] Orr, G. B. and Müller, K.-R. (2003). Neural networks: tricks of the trade. Springer. [76] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359. [77] Parkhi, O. M., Vedaldi, A., Zisserman, A., et al. (2015). Deep face recognition. In BMVC, volume 1, page 6. [78] Paulin, M., Revaud, J., Harchaoui, Z., Perronnin, F., and Schmid, C. (2014). Trans- formation pursuit for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3646–3653. [79] Qian, Q., Jin, R., Zhu, S., and Lin, Y. (2015). Fine-grained visual categorization via multi-stage metric learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3716–3724. [80] Quattoni, A. and Torralba, A. (2009). Recognizing indoor scenes. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 413–420. IEEE. [81] Ren, S., He, K., Girshick, R., and Sun, J. (2015a). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99. [82] Ren, S., He, K., Girshick, R., and Sun, J. (2015b). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS). References 81 [83] Rippel, O., Paluri, M., Dollar, P., and Bourdev, L. (2016). Metric learning with adaptive density discrimination. stat, 1050:2. [84] Rodriguez, A. and Laio, A. (2014). Clustering by fast search and find of density peaks. Science, 344(6191):1492–1496. [85] Roy, A. and Todorovic, S. (2017). Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3529–3538. [86] 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. International Journal of Computer Vision (IJCV), 115(3):211–252. [87] 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. International Journal of Computer Vision, 115(3):211–252. [88] Saleh, F., Akbarian, M. S. A., Salzmann, M., Petersson, L., Gould, S., and Alvarez, J. M. (2016). Built-in foreground/background prior for weakly-supervised semantic segmentation. In European Conference on Computer Vision, pages 413–432. Springer. [89] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823. [90] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off- the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 806–813. [91] Shi, H., Yang, Y., Zhu, X., Liao, S., Lei, Z., Zheng, W., and Li, S. Z. (2016). Embedding deep metric for person re-identification: A study against large variations. In European Conference on Computer Vision, pages 732–748. Springer. [92] Shimoda, W. and Yanai, K. (2016). Distinct class-specific saliency maps for weakly supervised semantic segmentation. In European Conference on Computer Vision, pages 218–234. Springer. [93] Simo-Serra, E., Trulls, E., Ferraz, L.,

Kokkinos, I., Fua, P., and Moreno-Noguer, F. (2015). Discriminative learning of deep convolutional feature point descriptors. In Computer Vision (ICCV), 2015 IEEE International Conference on, pages 118–126. IEEE. [94] Simon, M. and Rodner, E. (2015). Neural activation constellations: Unsupervised part model discovery with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 1143–1151. [95] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large- scale image recognition. CoRR, abs/1409.1556. 82 References [96] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE. [97] Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In Advances in Neural Information Processing Systems, pages 1857–1865. [98] Song, H. O., Xiang, Y., Jegelka, S., and Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, pages 4004–4012. IEEE. [99] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1):1929–1958. [100] Sun, L., Huo, Q., Jia, W., and Chen, K. (2015). A robust approach for text detection from natural scene images. Pattern Recognition, 48(9): 2906–2920. [101] Sun, Y., Chen, Y., Wang, X., and Tang, X. (2014). Deep learning face representation by joint identification-verification. In Advances in neural information processing systems, pages 1988–1996. [102] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015a). Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition. [103] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., et al. (2015b). Going deeper with convolutions. Cvpr. [104] Tang, P., Wang, X., Bai, X., and Liu, W. (2017). Multiple instance detection network with online instance classifier refinement. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [105] Tao, R., Gavves, E., and Smeulders, A. W. (2016). Siamese instance search for tracking. In Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, pages 1420–1429. IEEE. [106] Tsagkatakis, G. and Savakis, A. (2011). Online distance metric learning for object tracking. IEEE Transactions on Circuits and Systems for Video Technology, 21(12):1810– 1821. [107] Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In Proceedings of the IEEE International Conference on Computer Vision, pages 4068–4076. [108] Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. International journal of computer vision, 104(2):154–171. [109] Ustinova, E. and Lempitsky, V. (2016). Learning deep embeddings with histogram loss. In Advances in Neural Information Processing Systems, pages 4170–4178. References 83 [110] van Lint, J. H. and Wilson, R. M. (2001). A course in combinatorics. Cambridge university press. [111] Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011). The caltech- ucsd birds-200-2011 dataset. [112] Waltner, G., Opitz, M., and Bischof, H. (2016). Bacon: Building a classifier from only n samples. Proc. CVWW, 1. [113] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 3360–3367. IEEE. [114] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. (2016). Cnn-rnn: A unified framework for multi-label image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2285–2294. [115] Wang, Z., Chen, T., Li, G., Xu, R., and Lin, L. (2017). Multi-label image recognition by

recurrently discovering attentional regions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 464–472. [116] Wen, Y., Zhang, K., Li, Z., and Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In European Conference on Computer Vision, pages 499–515. Springer. [117] Wohlhart, P. and Lepetit, V. (2015). Learning descriptors for object recognition and 3d pose estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3109–3118. [118] Wu, C.-Y., Manmatha, R., Smola, A. J., and Krahenbuhl, P. (2017a). Sampling matters in deep embedding learning. In The IEEE International Conference on Computer Vision (ICCV). [119] Wu, C.-Y., Manmatha, R., Smola, A. J., and Krähenbühl, P. (2017b). Sampling matters in deep embedding learning. arXiv preprint arXiv:1706.07567. [120] Wu, R., Wang, B., Wang, W., and Yu, Y. (2015). Harvesting discriminative meta objects with deep cnn features for scene classification. In Proceedings of the IEEE International Conference on Computer Vision, pages 1287–1295. [121] Xie, S., Yang, T., Wang, X., and Lin, Y. (2015). Hyper-class augmented and regularized deep learning for fine-grained image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2645–2654. [122] Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. (2007). Multi-task learning for classification with dirichlet process priors. Journal of Machine Learning Research, 8(Jan):35–63. [123] Yi, D., Lei, Z., Liao, S., and Li, S. Z. (2014). Learning face representation from scratch. arXiv preprint arXiv:1411.7923. 84 References [124] Yoo, D., Park, S., Lee, J.-Y., and So Kweon, I. (2015). Multi-scale pyramid pooling for deep convolutional representation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 71–80. [125] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In Advances in neural information processing systems, pages 3320–3328. [126] Yuan, Y., Yang, K., and Zhang, C. (2017). Hard-aware deeply cascaded embedding. In The IEEE International Conference on Computer Vision (ICCV). [127] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In European Conference on Computer Vision, pages 818–833. Springer. [128] Zhang, J., Lin, Z., Brandt, J., Shen, X., and Sclaroff, S. (2016a). Top-down neural attention by excitation backprop. In European Conference on Computer Vision, pages 543–559. Springer. [129] Zhang, J., Wu, Q., Shen, C., Zhang, J., and Lu, J. (2016b). Multi-label image classifi- cation with regional latent semantic dependencies. arXiv preprint arXiv:1612.01082. [130] Zheng, L., Yang, Y., and Hauptmann, A. G. (2016). Person re-identification: Past, present and future. arXiv preprint arXiv:1610.02984. [131] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In Advances in neural information processing systems, pages 487–495. [132] Zhu, F., Li, H., Ouyang, W., Yu, N., and Wang, X. (2017). Learning spatial regular- ization with image-level supervisions for multi-label image classification. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [133] Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In European Conference on Computer Vision, pages 391–405. Springer. Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Borrowing Treasures from the Wealthy: Deep Transfer Learning through Selective Joint Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Multi-Evidence Filtering and Fusion

for Multi-Label Classification, Object Detection and Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and