Common visual pattern discovery and analysis

Wang, Zhenzhen

2021

Wang, Z. (2021). Common visual pattern discovery and analysis. Doctoral thesis, Nanyang Technological University, Singapore. <https://hdl.handle.net/10356/146898>

<https://hdl.handle.net/10356/146898>

<https://doi.org/10.32657/10356/146898>

This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0).

Downloaded on 26 Apr 2023 11:38:35 SGT

Common Visual Pattern Discovery and Analysis

Wang Zhenzhen

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

25 - 08 - 20

.....

Date

Wang Zhenzhen

.....

Wang Zhenzhen

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

26/8/20
[Enter Date in dd-mm-yy]
.....

Date

[Input Supervisor's Signature Here.]
.....

Prof. Tan Yap-Peng

Authorship Attribution Statement

This thesis contains material from four papers published in the following peer-reviewed journals / from papers accepted at conferences in which I am listed as an author, and also from one work which is under review.

Part of the Chapter 1 and Chapter 2 is from our survey paper published as:

Z. Wang, J. Meng, T. Yu and J. Yuan, “*Common visual pattern discovery and search*,” 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, 2017, pp. 1011-1018, doi: 10.1109/APSIPA.2017.8282178 [1].

The contributions of the co-authors are as follows:

- Prof. Yuan proposed the research direction.
- Prof. Yuan, Prof. Meng, Dr. Yu and I joined the discussion and narrowed the surveyed topics.
- I listed and summarized related publications and prepared the manuscript.
- Prof. Yuan, Prof. Meng and Dr. Yu revised the manuscript.

Chapter 3 is published as:

Z. Wang and J. Yuan, “*Simultaneously Discovering and Localizing Common Objects in Wild Images*,” in IEEE Transactions on Image Processing, vol. 27, no. 9, pp. 4503-4515, Sept. 2018, doi: 10.1109/TIP.2018.2839901 [2].

Z. Wang and J. Yuan, “*Unsupervised Multiple-Instance Learning for Instance Search*,” 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, 2018, pp. 1-6, doi: 10.1109/ICME.2018.8486471 [3].

The contributions of the co-authors are as follows:

- Prof. Yuan provided the initial project direction.
- Prof. Yuan and I discussed the challenges and possible solutions.
- I implemented the algorithms and performed all the experiments.
- I prepared the manuscript drafts. The manuscripts were revised by Prof. Yuan.

Chapter 4 is under reviewing:

Z. Wang, C. Xu, Y. Tan and J. Yuan, “*Attention-Aware Label Noise Learning for Image Classification*,” arXiv preprint arXiv:2009.14757 (2020) [4].

The contributions of the co-authors are as follows:

- Prof. Tan and Prof. Yuan provided the initial project direction.
- Prof. Tan, Prof. Yuan, Prof. Xu and I joined discussions on motivations and possible solutions.
- I implemented the algorithms, prepared data and performed all the experiments.
- I prepared the manuscript drafts. The manuscripts were revised by Prof. Tan, Prof. Yuan and Prof. Xu.

Chapter 5 is published as:

Z. Wang, W. Hong, Y. Tan and J. Yuan, “*Pruning 3D Filters for Accelerating 3D ConvNets*,” in IEEE Transactions on Multimedia, doi: 10.1109/TMM.2019.2950523 [5].

The contributions of the co-authors are as follows:

- Prof. Tan and Prof. Yuan provided the initial project direction.
- Prof. Tan, Prof. Yuan, Mr. Hong and I joined discussions of the project, including challenges, drawbacks of current solutions, our possible solutions.
- Mr. Hong and I discussed the details of the proposed algorithms.
- I implemented our proposed methods and conducted all the experiments.
- I prepared the manuscript drafts. The manuscript was revised by Prof. Tan, Prof. Yuan and Mr. Hong.

25 - 08 - 20

.....

Date

Wang Zhenzhen

.....

Wang Zhenzhen

Acknowledgements

First and foremost, I would like to express my sincere gratitude to Prof. Yuan Junsong, whose support is critical for me to be enrolled in NTU, and which is critical to both my career and my life. During my Ph.D study, he is always supportive and patient, encouraging me to explore the research topic I am interested. He is professional and knowledgeable, providing a lot of valuable suggestions and guidance to help me in all the time of research and writing of this thesis.

My sincere appreciation also goes to Prof. Tan Yap-Peng for his patience and motivation, for his support of my Ph.D study and my decision as attachment and intern. His insightful views and precious suggestions on both my research and future career broaden my horizons from various perspectives. I am deeply impressed by his professional attitude towards research and teaching, which continuously incites me to be such a person.

I feel grateful for being surrounded by so many friends in the past four years. The *Winners* Team, Weixiang, Tan, Jiong, Ze, Hu, and my dear friends, Mengjia and Yi, thank you. Although we may get lost in the journey of life, I will always remember the joyful moments we shared. Wish all of you a happy and wonderful life. I would also like to thank Yujun and my fellow labmates at ROSE Lab, Media Lab, and Visual Computing Lab for their help and support.

Last but not the least, I would like to thank my parents, older brother and sister-in-law for their support and love, and my little niece, Zihan and nephew, Hanbin, who are always proud of me. I should have been with them now if not the COVID-19, hope the world could go back to normal soon.

To *myself*.

Abstract

Given a set of images, common pattern discovery aims to explore the re-occurring patterns based on their similarities and differences. It is a long-standing but challenging task. On the one hand, patterns are frequently occurring visual primitives, and they are present in various forms, such as local features, semantic visual parts or visual objects. On the other hand, there exhibit large variations in visual appearances and structures even within the same kind of visual patterns which are exacerbated by prevalence of mobile-captured images. However, to distinguish visual patterns from one another is fundamental to many tasks in computer vision, such as pattern recognition/classification, object detection/localization, content-based image search. This thesis centers on task-driven common pattern discovery problems and designs several methods based on the characteristics of each task.

In the past decades, many studies have attempted to address the problem of visual pattern discovery. Most of them depend on hand-crafted feature representations and step-by-step hierarchically designed strategies, which are difficult to replicate and lack of generalization capability. Thanks to the development of deep learning, many computer vision tasks have evolved to a recorded high accuracy, and the convolutional neural network (CNN) has been involved either as a tool to generate discriminative features or as an end-to-end (e.g., image-to-label) learning model. However, the overwhelming performances are heavily dependent on large-scale high-quality labeled data that are costly to collect. In addition, over-parameterized CNN models which could consume huge computational resources are also critical for the impressive performances. Thus, in this thesis we make efforts to explore efficient ways to combine CNNs and common pattern discovery to address the problems in both tasks.

Particularly, we apply the common pattern discovery to three different tasks with different levels of combinations with CNNs:

- **Discovering and Localizing Common Objects:** Given a collection of unlabeled images, the goal is to simultaneously discover images that contain common objects (frequently appeared) and locate these objects in corresponding images. It has extensive practical usage, such as image summarization, user interests analysis from web images, etc.
- **CNN-based Noisy Label Learning:** The goal is to train a CNN-based classifier from images that may be incorrectly labeled, such as images from Flickr, instead of relying on the well annotated images which are costly to obtain. Although some images may be incorrectly labeled, the pattern contained in them could tell the truth. Based on the similarities to common patterns of each category, we could identify the wrongly labeled images and relabel them with the content-similar category. A new CNN layer and a general recursive training procedure are then proposed to learn from the refined image set.
- **3D CNNs Pruning:** Filters in CNN models also show patterns. There are thousands of weights in a typical CNN model which are learned in the form of filters. The over-parameterized CNNs consume large storage and computation, limiting their application to mobile devices and real-time systems. Hence, removing the redundant filters to accelerate models has attracted much attention and effort. We focus on the less studied 3D CNNs and present a novel algorithm, which treats filters as visual primitives and to distinguish the more informative filters from the redundant ones. Our method can also be generalized to any 2D CNN models without significant re-work.

In short, this thesis explores the common pattern discovery and applies it to solve various real-world problems. Several task-driven pattern discovery methods are proposed, along with the frontier strategy, Convolutional Neural Networks (CNNs), being well utilized and combined. Extensive experiments are conducted to evaluate the proposed methods and to make comparison with the state-of-the-art studies.

Contents

Acknowledgements	v
Abstract	vii
List of Figures	xii
List of Tables	xvi
Symbols and Acronyms	xvii
1 Introduction	1
1.1 Background and Motivation	1
1.2 Objective and Research Scope	6
1.2.1 Discovering and Localizing Common Objects	6
1.2.2 Learning from Noisy Images	8
1.2.3 Filter-based 3D CNNs Pruning	9
1.3 Major Contributions	10
1.4 Outline of the Thesis	12
2 Literature Review	13
2.1 Common Pattern Discovery	13
2.1.1 Local Visual Primitives based Pattern Discovery	14
2.1.2 Object Proposal based Pattern Discovery	14
2.2 Discovering and Localizing Common Objects	15
2.3 CNNs Learning from Noisy Images	17
2.3.1 Learning with Noisy labels	17
2.3.2 CNN Model Compression	20
2.4 Filter-based 3D CNNs Pruning	21
3 Simultaneously Discovering and Localizing Common Objects in Wild Images	24
3.1 Introduction	24
3.2 Methods	26
3.2.1 Simultaneous Object Discovery and Localization	26

3.2.2	Optimization	30
3.2.3	Ablation Simulation	32
3.2.4	Applications to Real-world Images	34
3.3	Unsupervised Object Discovery Experiments	35
3.3.1	The Object Discovery dataset	37
3.3.2	PASCAL VOC 2007 dataset	40
3.3.3	Discussion	44
3.4	Image Retrieval Experiments	45
3.4.1	Holidays	46
3.4.2	Oxford5k	47
3.5	Conclusion	50
4	Attention-Aware Noisy Label Learning for Image Classification	51
4.1	Introduction	51
4.2	Noisy Labels Purifying	52
4.2.1	Common Pattern Mining from Noisy Images	53
4.2.2	Noisy Image Relabeling and Purifying	53
4.3	Attention-Aware Noise Learning	54
4.3.1	Noise-Attention Model	55
4.3.2	Recursive Learning	58
4.3.3	Application to multi-attribute classification	61
4.4	Experiments	62
4.4.1	Manually Flipped Noise	64
4.4.2	Real-world Clothing Noise	70
4.5	Conclusion	72
5	Pruning 3D Filters for Accelerating 3D ConvNets	74
5.1	Introduction	74
5.2	Our Method	75
5.2.1	Filter Pruning via Determinantal Point Processes	76
5.2.2	Constructing DPPs Marginal Matrix L	78
5.2.3	Multiple Layers Pruning	80
5.3	Experiments	82
5.3.1	3D ConvNets Pruning	83
5.3.2	2D ConvNets Pruning	89
5.4	Conclusion	93
6	Conclusion and Future Work	94
6.1	Conclusion	94
6.2	Future Work	96
6.2.1	CNN Pruning from Feature Maps	96
6.2.2	Video Summarization	97
6.2.3	Diverse Image Retrieval	98

List of Author's Publications	101
Bibliography	103

List of Figures

1.1	The goal of common pattern discovery is to mine the frequently occurring visual primitives from a collection of images.	2
1.2	Examples of the visual primitives.	3
1.3	Visualization of CNN filters of AlexNet [6]. Left : the first Conv. layer, Right : the 2nd Conv. layer.	6
3.1	Our goal is to simultaneously discover images that contain common objects and also localize common objects in corresponding images. (Clearer when viewed in color.)	25
3.2	Schematic process of the key working mechanism. Given a set of images, we first extract the object proposals and then these proposals are filtered by a pretrained two-class classifier [7] which can better distinguish the objects from backgrounds. The weighted combination of the proposal's confidence output from the classifier together with the average saliency within the proposal is the final probability of the proposals containing an object. Then we build a densely connected graph which takes the scored proposals as nodes, and the similarities between these object proposals as the edges. Two models are proposed to handle different scenarios where 1) a given image group contains multiple common object classes, but each image contains at most one target object; 2) an individual image and the image group are both likely to contain multiple common objects. We only show the results under the first assumption in this figure; the common objects are highlighted in green and the outlier images are in red. (Clear when viewed in color.)	26

3.3	Illustration of sub-graph mining from a densely connected graph. The nodes of the graph are represented by circles in different sizes to indicate the confidence of proposals containing common objects. The different numbers in circles denote the proposals are extracted from different images. The edges of the graph are represented by the distances between circles, the more similar two proposals are, the closer they are in the figure. Each node is connected to the nearest 5 neighbors. The resulting sub-set of nodes are highlighted by green which enjoying high confidences and are cohesive with each other. Although close to each other, the red nodes are unlikely to contain object due to low weights of these nodes, and some of the grey ones in spite of enjoy high confidences but are unlikely to be common objects since they are unique to other nodes. (Viewed in color with magnification.)	29
3.4	The simulation results using <i>M-UOL</i> Model. Top: Groundtruth, and results using only node confidence, edge similarity and both. Middle: Results vary in terms of different values of η with the optimal λ (grid-search on ranges of values). Bottom: Results vary in terms of different values of λ with η being fixed as 0.3. (Better viewed in color.)	33
3.5	CorLoc(%) for the Object Discovery dataset with outliers using our <i>S-UOL</i> Model.	39
3.6	Examples of successful co-localization results for the Object Discovery subset in mixed-class setting. Left: Results using our <i>S-UOL</i> Model; Right: Outlier/Negative images.	40
3.7	Examples of co-localization for the PASCAL07-6 $\times$ 2 using the <i>S-UOL</i> Model under mixed-class setting.	41
3.8	Examples of co-localization for the PASCAL07-all using the <i>S-UOL</i> Model in mixed-class setting, we resize the ratio of width and height in order to show elegantly.	43
3.9	Samples of the two datasets, one for image retrieval and another for instance retrieval.	46
3.10	Examples of instance retrieval results on Oxford5k.	49
4.1	Examples of noisy labels. The image annotations that are manually labeled by amateurs or automatically generated by a machine are not reliable. Worse, the noisy labels and the number of mislabeled images are not specified.	52
4.2	An overview of the proposed attention-aware noisy learning approach. Firstly, a Noise-Attention model is introduced, which consists of multiple noise-specific units (NS unit) with each paying attention to a specific noisy distribution. The attention network is trained using the proposed recursive learning strategy. At each recursive round, the training supervisions combined from the given labels and the outputs from the previous round will guide the network learning in the current round.	55

4.3	Illustration the definition of the noise-specific unit.	56
4.4	Cloth samples with multiple attributes. The first row shows samples from training set (red ones are mislabeled), and second row from testing set.	61
4.5	Test errors (%) on CIFAR-10 and SVHN in terms of number of Noise-Specific Units in Noise-Attention model.	66
4.6	Test errors (%) on CIFAR-10 and SVHN in terms of number of recursive rounds.	67
4.7	Visualization of noise-specific units. The first column represents $Q_1 = I$, the rest ones show the learned noise-specific units.	67
4.8	Test errors (%) on real-world cloth dataset in terms of number of noise-specific units in Noise-Attention model and number of recursive rounds.	71
5.1	Filter pruning in 2D ConvNets has shown excellent performance due to the redundant parameters in ConvNets, even though the inputs for 2D ConvNets are unrelated images. Will 3D ConvNets, whose inputs are sequential frames, be even more redundant?	75
5.2	Illustration of our proposed 3D filter pruning. Given a well-trained 3D ConvNets, the duplicate and less informative filters are pruned (highlighted by gray in $\mathcal{F}_\ell$). Consequently, the following feature maps can also be removed, so can the filter cubes corresponding to the removed feature maps.	76
5.3	Visualization the distribution of importance score $\mathbf{q}_i$ of filters in all convolutional layers of C3D trained on Sports1M.	79
5.4	Maximum Abs of Filters (MAF) representation. The filter cube with the maximal absolute values are concatenated as the compact filter representation Φ_i	80
5.5	Filter pruning across multiple layers. The filters pruned in the ℓ -th layer (green by column) will affect the number of feature maps generated by them and also the size of filters in $(\ell+1)$ -th layer (green by row).	81
5.6	C3D (Top) and P3D (Bottom) network on Sports1M: Single layer performance analysis under different pruning ratios (w/o retraining).	84
5.7	Visualization of filters in conv1 of C3D network trained on Sports1M, with about 20% (12/64) pruning ratio. Filters are ranked by its importance score $\mathbf{q}_i$ in a descending order from top left to bottom right. Filters with red boxes are the ones pruned by our method. The last 12 filters circled by black dashed lines are the ones pruned according to criteria in [8], i.e., pruned the ones with least ℓ_1 -norm.	85
5.8	VGG-16 network on CIFAR-10 (Top) and ImageNet (Bottom): Single layer performance analysis with different pruning ratios (w/o retraining).	90
6.1	Visualization of feature maps output from a CNN layer (Left) and the corresponding patches in the original images (Right).	96

6.2	Examples of diverse image retrieval.	98
-----	--	----

List of Tables

3.1	Comparison of different supervisions required for the Weakly-/Un-supervised methods and ours.	25
3.2	CorLoc(%) for the Object Discovery subset using <i>S-UOL</i> Model.	38
3.3	F1-scores for the Object Discovery subset using the <i>M-UOL</i> Model.	39
3.4	CorLoc(%) for PASCAL07-6×2 using the <i>S-UOL</i> Model. The methods labeled by ‘*’ are in weakly supervised way.	41
3.5	Average CorLoc(%) for PASCAL07-6×2 using the <i>S-UOL</i> Model.	41
3.6	CorLoc(%) for PASCAL07-all using the <i>S-UOL</i> Model. The methods labeled by ‘*’ are in weakly supervised way.	42
3.7	Average F1-scores for the PASCAL07 using the <i>M-UOL</i> Model.	43
3.8	Comparison of different ways of selecting proposals.	44
3.9	Average CorLoc(%) in terms of various weights using S-UOL Model.	44
3.10	Comparison of CNN-based features with hand-crafted features.	44
3.11	The results of image retrieval on Holidays dataset.	47
3.12	The results of instance retrieval on Oxford5k.	48
4.1	Noisy levels (%) before and after relabelling.	65
4.2	Test errors in the presence of noisy labels.	68
4.3	The statistics of the real-world clothing dataset used in our experiments.	70
4.4	Test errors (%) of the considered methods on clothes images.	72
5.1	Statistics of baseline models.	82
5.2	Comparison of several baselines and our proposed on Sports1M dataset. Top: C3D. Bottom: P3D199.	86
5.3	Comparison of SOTA and our proposed method on Sports1M dataset. Top: C3D. Bottom: P3D199.	87
5.4	Comparisons of the proposed MAF to the naive vectorized filter representation (Top-1 Acc drop (%), with fine-tuning).	89
5.5	Comparisons of pruning results on Cifar10 in terms of various pruning ratios (PR).	91
5.6	Comparisons of pruning results on ImageNet in terms of various pruning ratios (PR).	92
6.1	Experimental results (mAP (%)) of our diverse image retrieval.	100

Symbols and Acronyms

Symbols

$\{\mathcal{I}_i\}_{i=1}^N$	a set of N samples
$\mathbf{f} \in \{0, 1\}^N$	a binary vector of length N
$A \in \mathbb{R}^{M \times N}$	a 2D matrix of size $M \times N$
$\ \cdot\ _n$	the n -norm of a vector
$\mathcal{O}(\cdot)$	the computational complexity
$\{x_n, y_n\}$	n -th sample x_n with label y_n in an image set
$W^\ell \in \mathbb{R}^{n_{\ell+1} \times n_\ell \times k \times k \times d}$	a 5D matrix to denote weights in a convolutional layer
$\det(\cdot)$	the determinant function

Acronyms

S-UOL	Single Instance of Unsupervised Object Localization
M-UOL	Multiple Instance of Unsupervised Object Localization
IoU	Intersection over Union in terms of two image patch
CorLoc	correct localization, the number of images correctly localized
MAC	maximum activation of a convolutional layer
MAF	maximum absolute value of filters
mAP	mean Average Precision
A^2NL	an attention-aware noise learning framework
FLOPs	floating point operations per second
DPPs	Determinantal Point Processes

Chapter 1

Introduction

1.1 Background and Motivation

With the prevalence of cameras and mobile devices, people take and upload large amounts of images every day. The big visual data era brings many computer vision opportunities and challenges. On the one hand, with large-scale labeled samples, many tasks including image/video classification, object detection and segmentation, just to name a few, have achieved remarkable progress. On the other hand, it is laborious and costly to manually annotate the huge number of images, not to mention that different annotations are required for different tasks. Unfortunately, most images presented online are usually unlabeled or with coarse or incorrect labels, making it difficult to directly use them in computer vision tasks. In such cases, the common pattern discovery, which mines the visual correlations from contents instead of labels, has received much attention and interest.

Common visual pattern discovery aims to explore the frequently appeared composition of visual primitives from a collection of visual data even without manually labeled annotations [9, 10]. Figure 1.1 illustrates the general procedure of common pattern discovery. As an unsupervised or weakly-supervised task, it could

FIGURE 1.1: The goal of common pattern discovery is to mine the frequently occurring visual primitives from a collection of images.

dramatically reduce the time and labor required to annotate visual data, especially in this big visual data era where there are tons of visual data being uploaded to Internet every day. This topic is also fundamental to many computer vision problems, such as image classification, content-based image retrieval, and object detection/localization, as the mined common patterns could help to perceive and analyze the given image collections. Based on the similarities of a specific pattern and the differences among different patterns, we can differentiate a dog from a car (classification/recognition), the foreground object from the background (detection/localization), or recognize the images containing desired patterns from an image pool (instance/image retrieval).

As a fundamental research topic, common pattern discovery could be qualitatively evaluated by viewing the mining results or be applied to computer vision tasks and thus it can be quantitatively evaluated by the results of the whole system. For the former, as the “common” is not a rigid description, it is not easy or even not necessary to evaluate or to compare with other methods, limiting its influence to peer researchers and to this field. Take a set of images containing airplanes

FIGURE 1.2: Examples of the visual primitives.

as an example, the sky, airplanes and its parts are all present very frequently; therefore mining results could be some or all of the patterns. In order to make a fair quantitative comparison with the literature, in this thesis, we focus on combining the common pattern discovery with real-world applications, and proposing several task-driven pattern discovery algorithms to address the problems at the frontier of computer vision.

However, to discover visual patterns from a random collection of images is quite a challenging task, in part because there is no rigid definition for visual primitive. For example, the visual primitives could be semantic visual parts as shown in Figure 1.2. A bicycle is composed of two wheels (circles) and one triangle skeleton; each part of the bicycle could be seen as an individual visual primitive [11, 12], while the whole bicycle itself can also be regarded as a visual primitive. Even with a predefined metric in a specific task, it is not easy to extract the good-quality visual primitives from unlabeled data. Another challenge is that the visual primitives can be very diverse on their own. Large variations may present in visual appearance and structure. In Figure 1.2(b), we can see that the bicycle wheels could vary largely, not to mention the whole bicycle. Various illuminations and scales, as well as the occlusions and distortions further exacerbate the difficulties. Therefore, feature representations with strongly discriminative ability are no-doubt critical.

Extracting visual primitives from image collections is the very first step for visual pattern mining, and good-quality visual primitives can definitely contribute to the mining results. The methods on collecting visual primitives can be roughly divided into approaches based on local feature and those based on semantic object

proposals. For the former, many local feature detectors are popularly used to obtain visual primitives, such as Blobs [13], Normalized Cuts [14], Deformable Primitive Models [15], etc. These methods usually depend on straightforward strategies to exhaustively decompose images, and they tend to generate superfluous number of visual primitives, leading to heavy computation cost. With the development of deep learning, especially deep Convolutional Neural Networks (CNNs), many methods on obtaining object proposals have been proposed, such as EdgeBoxes [16], DeepBoxes [7] and Bing [17]. Compared to the traditional methods, the CNN-based methods could predict a score for each candidate proposal, based on which the number of candidate segmentations can be significantly reduced by tuning the threshold.

The following step is to extract feature representations from the obtained visual primitives. The hand-crafted criteria, such as SIFT descriptor, Bag-of-Words, are usually resorted to by traditional pattern mining methods. The two-step strategy, from images to visual primitives then to features, involves manual operation and experience; therefore they are difficult to replicate and lack of generalization capability, and the performances are usually not good enough for practical problems, making them difficult for wide applications. In contrast, CNN models can take images as input, and simultaneously output the object proposals and their features by a one-step manner, and these features are proven to be more discriminative than hand-crafted ones in many computer vision applications. Thus, it is highly possible that the traditional pattern mining methods could be enhanced by better inputs provided by CNN models. In Chapter 3, we explore the effectiveness of using CNN models to preprocess visual data for the problem of common object discovery, and also introduce a new algorithm to make better use of them.

Except for being used in the preprocessing, CNN models are more frequently used as end-to-end solutions, and have achieved record-high accuracies in many computer vision tasks, triumphing over numerous traditional methods. A typical end-to-end CNN model could take images as inputs and directly output their labels, while traditional methods need to extract features first and then be used to train

the classifier. The end-to-end strategy is more likely to achieve global optimization and results in models with better performance. However, the overwhelming performance of CNNs is heavily dependent on large-scale data with high-quality labels e.g., ImageNet dataset [18], which are typically laborious and costly to be collected and annotated. In contrast, there are millions of freely available images with user-supplied labels that can be easily accessed from the web. Although the labels could be coarse or incorrect, learning from web images or noisy images is a promising solution to relieve the heavy dependence on finely-labeled visual data. To make use of the noisy data, in Chapter 4, we propose to firstly identify and re-label the noisy images by the inherent patterns of images rather than the observed labels. Based on the learned common patterns of a category, we are able to correct the images that are wrongly labeled to other categories or other images that are wrongly labeled to this category. We also propose a novel CNN layer and a general recursive training strategy for better utilizing the refined image set.

The superior performance of CNN models also credits to the over-parameterized models with wide and deep architectures, which consume large amount of memory to store and huge computational costs to run, limiting their wide usage in mobile devices and real-time scenarios. There are thousands of weights in a typical CNN model which are presented in the form of filters, and filters constitute layers. To accelerate the CNNs and reduce storage, researchers recently attempt to prune away the redundant or unimportant weights, filters or even layers. Compared with layer-level and weights-level pruning, filter-level pruning is appropriate in granularity and is a structured way without introducing the sparsity so that it does not require sparse libraries or any specialized hardware. In Chapter 5, by treating each filter as an visual primitive, we propose an offline filter-level method for optimizing 3D ConvNets from the view of common pattern discovery. The motivation of the proposed method is that filters show obvious patterns, as seen in Figure 1.3. The similar patterns suggest a pair of duplicate filters in functionality, therefore one of them could be pruned away without severe degrading the effect of the whole layer. The more filters in a CNN model show similar patterns, the

FIGURE 1.3: Visualization of CNN filters of AlexNet [6]. **Left:** the first Conv. layer, **Right:** the 2nd Conv. layer.

higher potential it could be pruned. Thus to discover the similar and common patterns/filters is key to recognize the redundancy and accelerate CNN models.

Common pattern discovery has many potential applications. In visual search, the targets are selected based on the pattern matching between query instances/images and the reference images. In object categorization, the intra-class common patterns are used to distinguish different categories. In video analysis, such as key-frame selection, the frequently appeared visual primitives are hints for redundant frames. Except for those conventional applications, in this thesis, we also explore the intersection of common pattern discovery and deep learning techniques, using deep learning tools to improve the pattern discovery performance, and using the idea of common pattern discovery to address the problems in deep learning area.

1.2 Objective and Research Scope

1.2.1 Discovering and Localizing Common Objects

Motivated by the recent success of supervised and weakly supervised common object discovery, we move one step further to tackle common object discovery in a

fully unsupervised way. Generally, object co-localization aims at simultaneously localizing objects of the same class across a group of images. Traditional object localization/detection usually trains specific object detectors which require bounding box annotations of object instances, or at least image-level labels to indicate the presence/absence of objects in an image. Given a collection of images without any annotations, we propose a fully unsupervised method in Chapter 3 to simultaneously discover images that contain common objects and also localize common objects in corresponding images. Without requiring to know the total number of common objects, we formulate this unsupervised object discovery as a sub-graph mining problem from a weighted graph of object proposals, where nodes correspond to object proposals and edges represent the similarities between neighbouring proposals. The positive images and common objects are jointly discovered by finding sub-graphs of strongly connected nodes, with each sub-graph capturing one object pattern. The optimization problem can be efficiently solved by the proposed maximal-flow-based algorithm.

Instead of assuming each image contains only one common object as in previous works, our proposed solution can better address wild images where each image may contain multiple common objects or even no common object. Moreover, our proposed method can be easily tailored to the task of image retrieval in which the nodes correspond to the similarity between query and reference images. Extensive experiments on PASCAL VOC 2007 and Object Discovery datasets demonstrate that even without any supervision, our approach can discover/localize common objects of various classes in the presence of scale, view point, appearance variation, and partial occlusions. We also conduct broad experiments on image retrieval benchmarks, Holidays and Oxford5k datasets, to show that our proposed method, which considers both the similarity between query and reference images and also similarities among reference images, can help improve the retrieval results significantly.

1.2.2 Learning from Noisy Images

Except for the conventional applications to categorization or image retrieval, common pattern discovery can also be creatively combined with deep learning techniques to solve the new problems emerged along with the rapid development of CNNs. The most impressive thing that CNNs bring to the field of computer vision is the overwhelming performance, which benefits from the large-scale labeled samples. However, obtaining high-quality annotations is laborious and costly. Then it naturally comes to the solution that uses images and user-supplied labels crawled from websites as a substitution. Existing methods on utilizing noisy data can roughly be divided into two ways, to improve the quality of training data or to design a robust model; we combine both. Our solution is proposed in terms of image classification, the goal is to learn a classifier from the noisy images that could be as good as that from the well-labeled images.

In Chapter 4, we propose a content-based strategy to refine the noisy image set based on the assumption that the number of correctly labeled images accounts for the majority in the given dataset. In particular, we apply the pattern mining method proposed in Chapter 3 to images with the same observed labels, where the mined common patterns work as the k -medoids, and k is the number of observed categories. Similarities between images and the mined common patterns are used as a criterion to relabel images.

Then an attention-aware noisy label learning approach (A^2NL) is proposed to improve the discriminative capability of the network trained on the refined dataset. Specifically, a Noise-Attention model, which contains multiple noise-specific units, is designed to better capture noisy labels. Each unit is expected to learn a specific noisy distribution for a subset of images so that different disturbances are more precisely modeled. Furthermore, a recursive learning process is introduced to enhance the learning ability of the attention network by taking advantage of the learned high-level knowledge. In the proposed recursive learning, by combining the given training labels and the mined knowledge in the previous iteration

as the training supervisions, a well-trained attention network is able to bootstrap itself in the current iteration. To fully evaluate the proposed method, we conduct experiments from two aspects: manually flipped label noise on large-scale image classification datasets, including CIFAR-10 and SVHN; and real-world label noise on an online crawled clothing dataset with multiple attributes. The superior results over state-of-the-art methods validate the effectiveness of our proposed approach.

1.2.3 Filter-based 3D CNNs Pruning

Filters in CNN models also show patterns; therefore they can be treated as visual primitives. In light of this, we apply the common pattern discovery to CNN models to analyze the importance of filters. Then the less important filters could be pruned away from the over-parameterized CNN models, leading to fast and light-weight models. The aim of the network pruning is not only to accelerate the running process of CNNs, but also to maintain the performance.

Many methods have been proposed to accelerate 2D ConvNets by removing redundant parameters. However, few efforts are devoted to the problem of accelerating 3D Convolutional Networks. The 3D ConvNets, which are mainly designed for extracting spatiotemporal features, have been widely used in many video analytics tasks, such as action recognition and scene analysis. In Chapter 5, we focus on accelerating 3D ConvNets for two motivations: (1) Fast video processing techniques are in dire need due to the explosive growth of video data; (2) Compared with individual images, video data consist of consecutively similar frames, so that they are inherently more redundant. We present a novel algorithm to dramatically accelerate 3D ConvNets by pruning redundant convolutional filters, while preserving the discriminative power of the networks. Specifically, we formulate the filter pruning from 3D ConvNets as a subset selection problem where each filter is regarded as a visual primitive candidate. Determinantal Point Processes (DPPs) are employed to discriminatively select the filter candidates which are informative and yet diverse. We evaluate our method using two popular 3D networks, C3D and

Pseudo-3D, on Sports-1M dataset for video classification. Extensive experimental results demonstrate both the efficiency and performance advantages of our method. We also show that the proposed method can be easily generalized to 2D ConvNets pruning with promising experimental results on VGGnet and ResNet.

1.3 Major Contributions

In this thesis, we propose several task-driven common pattern discovery methods and evaluate them with the results of the corresponding tasks. The contributions are summarized in terms of each sub-problem.

- **Discovering and Localizing Common Objects:** Motivated by the remarkable progress in object proposal generation, we treat object candidates as visual primitives and mine the frequently appeared objects from unlabeled real-world images. Compared to existing methods, our method could be used in a more complex and practical scenario. In particular, our contributions are three-folds:
 - (1) A new sub-graph selection objective function for unsupervised common object discovery is proposed to formulate the scenario where the number of objects contained in each image is unconstrained.
 - (2) A novel solution which is inspired by the maximal flow algorithm is proposed to optimize the sub-graph selection problem efficiently.
 - (3) A very first attempt to simultaneously perform common object discovery and localization with high outlier ratios (*e.g.*, as high as 60%) in a fully unsupervised situation.
- **Learning from Noisy Labels:** Using images with low-quality but easy to get annotations to substitute the well-annotated images has been a trend to train CNN models. However, training with the coarse or incorrect labels, dubbed noisy labels, will lead to suboptimal models. We improve the learning

ability of CNN models by mining the class patterns from the contents except for the label supervision. The contributions are summarized as follows:

- (1) A pre-processing strategy is proposed to improve the label quality of training data. Based on the similarities of images to all classes, we are able to correct some unreliable labels and relabel the images based on their content nearest neighbor.
- (2) A Noise-Attention model with multiple noise-specific units is proposed to mine different distributions of label noise, which can be flexibly extended to other standard CNNs.
- (3) A recursive learning strategy, which could assemble the high-level knowledge learned from multiple networks, is introduced to boost the attention network learning ability.
- (4) Extensive experiments on both manually flipped label noise and real-world label noise demonstrate the excellent effectiveness and robustness of our attention-aware noisy label learning framework.

- **Filter-based 3D CNNs Pruning:** In light of the increasing efforts on accelerate over-parameterized CNN models, we address this problem by treating filters as visual primitives, and propose a filter-level network pruning method. Our main contributions lie in follows:

- (1) Our proposed filter pruning method focuses on 3D ConvNets compression. Particularly, we formulate it as a subset selection problem by regarding each filter as a candidate.
- (2) We introduce Determinantal Point Processes (DPPs) as a flexible and powerful tool for modeling filter level importance and diversity. A practical method is also presented for creating DPPs over filters.
- (3) A filter representation, Maximum Abs. of Filters (MAF), is proposed for efficient similarity measurement, which is compact and translation invariant.

- (4) We validate the effectiveness of our proposed method by extensive experiments on accelerating two popular 3D networks for video classification and the generalization to 2D network.

1.4 Outline of the Thesis

The remaining of this thesis is organized as follows: In Chapter 2, we review the related work of common pattern discovery, both traditional methods and CNN-based methods, as well as their applications. In Chapter 3, we elaborate the proposed sub-graph mining methods and its optimization. Instantiation on real-world images for object localization and instance retrieval is also implemented. In Chapter 4, we combine the common pattern discovery with CNN learning from noisy labels, and propose methods to improve the learning ability in terms of the image classification task. In Chapter 5, we treat filters as visual primitives and formulate the network pruning problem as DPPs selection from filters. Although we target for 3D CNNs for video classification, the proposed method extends beyond 3D CNNs and also achieves competitive performance on 2D CNNs. Finally, Chapter 6 concludes previous chapters and presents potential and future research directions based on this thesis.

Chapter 2

Literature Review

2.1 Common Pattern Discovery

Common pattern discovery plays an important role in computer vision tasks, such as object detection/localization [19, 20], image co-segmentation/saliency [21–23]. The main paradigms of these tasks are similar: the inputs are usually real-world images with incomplete labels or sometimes even without any supervised information, then the key step is to discover the most frequently occurring patterns by methods such as local feature matching, sub-graph mining, etc. Based on the results of pattern discovery, the outputs differ according to the targets of tasks; for example, detection/localization draws bounding boxes around objects, while co-saliency and co-segmentation predict pixel-wise labels. According to different granularities, existing methods of common pattern discovery could be roughly grouped as local-feature based and object-proposal based; they are also distinguished by using CNN models or not.

2.1.1 Local Visual Primitives based Pattern Discovery

Many local feature detectors [13, 24] have been used to obtain visual primitives, such as blobs. Normalized Cuts [14], which is firstly proposed to solve the perceptual grouping problem for image segmentation, can also be used to collect primitive regions. The deformable primitive model [15], which is extensively used in object detection tasks, can be adopted to generate object primitives.

Then two kinds of approaches, bottom-up and top-down, have been used for mining common patterns from those visual primitives. For the former, different models are designed to gradually group the visual primitives extracted from image collections, then the frequent spatial compositions of visual primitives are selected as visual patterns. For the latter, the models are directly built on the labeled images and the segmentations to learn the parameters, so that the trained models could be used to infer the common visual patterns from unseen images.

2.1.2 Object Proposal based Pattern Discovery

Different from the methods based on local visual primitives which iteratively merge the image patches until the larger objects are found, recent studies tend to directly generate the patches from a collection of images that contain the whole objects, *i.e.*, the object proposals [25, 26]. There are many methods for obtaining object proposals, such as Selective Search [27], Randomized Prim's [28], EdgeBoxes [16], and Bing [17]. These methods usually generate the candidate proposals with scores indicating the probability of containing an object, so that the number of candidates can be significantly reduced compared to the dense framework, *e.g.* sliding window.

After obtaining the object proposals, many classic methods designed for pattern discovery from visual primitives, such as sub-graph mining [29–31], K-medoids clustering [32, 33], and dense correspondence [34], could be applied to common objects discovery. In recent years, CNN-based methods have been popular. Some methods are used in a supervised end-to-end manner, others are unsupervised,

such as autoencoder [35]. And both of them achieve competitive or even better performance than traditional methods.

2.2 Discovering and Localizing Common Objects

As a special case of common pattern discovery, common object discovery aims to find the frequently appeared objects from an image sets, which are of course the common patterns. In the following, we will review some representative studies according to different levels of supervision required, and also their extension to image/instance retrieval, then the strategies of the sub-graph mining which result in a novel solution to the proposed method in Chapter 3.

Weakly-supervised Object Discovery. Weakly-supervised strategies are widely used in the tasks such as object detection/localization [19, 20, 36], image co-segmentation/saliency [22, 37, 38]. Object discovery from videos [39–41] can also be seen as weakly-supervised methods since we only know the frame-level labels. Although many studies claim to be weakly-supervised methods, the supervision required may not be the same. Some methods require the image-level labels to indicate the categories, while some only need the binary labels to indicate whether there exists an object or not. Regardless which methods, obtaining the image-level labels is much easier than annotating object locations or pixel-wise labels. Some studies [42, 43] on mid-level representations also adopt the weakly-supervised skill by requiring only the image labels rather than the patch labels, so that they can leverage the large amount of tagged images on the Internet.

Unsupervised Object Discovery. More recently, researchers tend to study object discovery tasks in wild images without any annotations, including noisy images without target objects and images containing multiple objects. The most representative studies on common object co-localization are [44–46]. These methods together with our proposed method take advantage of the object proposals. However, these methods are based on the assumption that there is only one target object

in each positive image, and the image groups are usually without any outliers or with a rather low outlier ratio. Since these restrictions are not always satisfied in real-world image sets, we extend our proposed method to tackle scenarios where individual images could contain multiple target objects, and also scenarios with high outlier ratios (*e.g.*, as high as 60%). Clustering-based methods [47, 48] are also well studied for classification or co-segmentation, they can also be used as a post-processing of the co-localization to group the patches in the same category.

Object Instance/Image Search. The goal of image/instance retrieval is to select the images, which are similar to or contain the same instance as the given query, from a set of reference images. The difference of image retrieval and instance retrieval is that the former prefers to select images that are globally similar to the query [49, 50], while the latter cares more about the foreground objects contained in the reference images [33, 51]. The resulting ranks of the two tasks both depend on the similarity to the query, and usually there is a re-ranking stage to refine the search results, such as AML [51] and QE [52]. However, it is widely acknowledged that if a reference image I is associated with a query image I_q , then it is possible that the reference images which enjoy high similarity with I may also be associated with I_q , and vice versa. Thus, our proposed framework in Chapter 3, which employs both the similarity between query and reference images and the similarities among reference images, can help to improve the resulting ranks by adding or removing some outlier images from the top-ranked list.

Sub-graph Selection. Frequent feature selection with graph [53, 54] is one of main topic of data mining. It has long been used in computer vision for common object/pattern discovery. For example, the methods of sub-graph mining have been introduced to solve the task of thematic pattern discovery in videos [55, 56], where each node of the graph is represented by visual words and the affinity graph is built based on the relationships of these visual words. The sub-set of visual words having the maximum overall mutual information scores is selected. Although the algorithms work well in these scenarios, to find a sub-graph only depends on graph edges but neglects the weights of nodes is obviously inadequate for our settings.

For instance, given a group of images containing airplane, the patches containing sky can also be the frequently occurring sub-sets since the airplane usually occurs together with sky. In order to resolve such confusion, in Chapter 3, we propose to apply prior knowledge like saliency, ratio of the width to the height to discriminate the foreground objects from backgrounds if they have similar frequency.

2.3 CNNs Learning from Noisy Images

In image classification task, there are mainly two kinds of noise in a dataset, feature noise and label noise. The former, also known as image noise, corresponds to corruption in observed features of data, which can range from random variation of brightness or color information in the images captured, such as Gaussian noise and salt-and-pepper noise. Label noise is a common but complex issue, it refers to the observed labels that associated to instances might be incorrect, e.g., a truck is labeled as train and a furry dog is labeled as cat, which is inevitable due to imperfect evidence, patterns or insufficient information for reliable labeling. Both noises try to cause a significant decrease in performance but label noise attracts more attention in recent years. On the one hand, there have been many methods introduced to address the feature noise, from traditional methods such as frequency-dependent noise reduction to CNN-based attention mechanism, making the feature noise not a significant limitation anymore. On the other hand, the label is unique for each data and has a significant impact not only on the sample itself, but also on the classifiers, especially the CNN-based ones. Therefore, in this thesis, we focus on the label noise and propose an end-to-end CNN model to relieve its influence on the image classification.

2.3.1 Learning with Noisy labels

The common setting of noisy label classification and the weak-/semi-supervised learning problem is that only a part of correct labels is provided during training,

while the main difference is that those high-quality samples are specified in the latter. In noisy label classification, we do not know which samples are incorrectly labeled, neither the total number of mislabeled samples, which make the problem more challenging than weak-/semi-supervised problem. Generally, the algorithms designed for noisy label classification also could be used for weak-/semi-supervised problems by randomly annotating the unlabeled samples, but not vice versa.

Label noise comes from various sources and is presented with different types. It not only degrades the classification performance and the stability of feature rankings, but also increases the complexity of the model and the number of instances needed to achieve results competitive with that of noise-free datasets. The extreme practice is not to use all supervisory labels. For example, instead of using traditional classification-based methods to learn image representations, Cui *et al.* [57] build an un-/semi-supervised CNN model that could take advantage of semantic prior knowledge and relations among image contents to learn the image embedding without using labels. Kyunghyun *et al.* [58] resort to the encoder-decoder networks to get rid of the dependence on supervisions. However, such a solution also abandons the useful information contained in supervisory labels. In this work, we narrow the discussion on methods that wisely utilize the noisy labels, which we roughly classify into three categories, and discuss the recent and representative works that are highly related to our proposed method.

Noisy Samples Cleaning. The most straightforward way to handle label noise is to improve the quality of image collections by label noise cleansing skills [59–63]. For example, [60] employs a Bayesian classifier to evaluate the probabilities of an instance belonging to all considered class labels, and using the entropy calculated from the probability distributions to decide whether the instance is mislabeled or not. The instances predicted as wrongly labeled will be relabeled as the predicted class. Jeatrakul *et al.* [62] learn a Neural Network using training data and remove all instances that are misclassified by the Network. Recently, Wang *et al.* [63] propose a recursive learning schema, where a Siamese network is employed to learn discriminative features from predicted clean and noisy instances by a noisy detector.

Alternatively, the learned discriminative features can be used to train the noise detector. However, these methods suffer from a contradictory dilemma, i.e., label noise datasets often lead to poor classifiers, whereas high-quality classifiers are necessary for precise predictions. Thus removing or relabeling mislabeled instances may accumulate noises.

Robust Loss Designing. To build models that are intuitively robust to label noise is another solution. In [64], a label noise robust logistic regression model is proposed to estimate the label flipping probabilities and to tackle high dimensional noisy settings. Bootkrajang and Kaban [65] derive two noise robust approaches from AdaBoost. One is to directly build a label-noise robust classifier, the other is to design a noise-specific loss function to improve the robustness of the proposed AdaBoost model. Semi-supervised learning [66] is another popular algorithm to address label noise. Duan *et al.* [67] introduce a propagation algorithm to iteratively predict noisy and reliable images, which are then used to refine the training data and to re-train the classifier. To avoid propagation of mislabels, they set a strict constraint on labels that can be propagated. Zero shot learning [68] can also be employed to handle datasets with mislabeled or unlabeled instances. In [68], Jayaraman and Grauman presented a novel random forest approach which can be used to leverage statistics about the error tendencies of labels. Pan *et al.* [69] propose the robust latent poisson deconvolution for topical detection from noisy web images. However, the main problem of the aforementioned approaches is that they increase the complexity of learning algorithms and can lead to overfitting.

Noisy Distributions Modeling. To directly model the noisy distributions of the given image collections is another approach to address the challenges. In [70], Sukhbaatar *et al.* model the noisy distribution by a fully-connected layer. Based on the assumption that all instances from a dataset following the same noisy distribution, they add an extra noise layer on top of the softmax layer to adapt the network outputs to match the noisy label distribution. During training, all the instances will pass through the noise layer to re-weight their confidence in each class. Inspired by them, Xiao *et al.* [71] build a probabilistic graphical model to learn the

relationships between images, class labels and label noise. Besides, the method also introduces an explicit indicator of z to denote the noisy type. Although achieving improvement over [70] to some extent, the model needs twice the number of parameters as two networks are combined during training and inference. In [72], Jacob and Ehud model the noise by an additional softmax layer which connects the correct labels to the noisy ones. Unlike [70] which uses SGD to optimize the whole network, [72] utilizes the EM algorithm to optimize the noisy-label likelihood function. Our proposed method in Chapter 4 falls into this category with goals to learn the noisy distributions. But different from previous methods, we propose a Noise-Attention model containing multiple noise-specific units to cater for different noisy distributions. In addition, a recursive learning strategy is designed to boost the learning ability of the proposed noise-attention network.

2.3.2 CNN Model Compression

It is well-known that ensemble methods can be used to improve prediction performance [73]. A straight forward way is to combine the predictions of multiple models by weighted averaging or voting [74–76]. However, the large, complex ensembles will consume huge computational and storage resources. To compress the ensembles into smaller, faster models, a method for model compression is proposed in [77]. Specifically, Bucilă *et al.* focus on training compact artificial neural nets to mimic the function learned by ensemble selection. To achieve this goal, the combined predictions of the ensemble are used as supervisions to train the network, so that it could make predictions similar to the ensemble. By this way, the knowledge in an ensemble is possible to be compressed into a single model which is much easier to deploy. Hinton *et al.* develop this approach further to Deep Neural Networks (CNNs) [78] which is featured by a softmax layer on the top. In the work, a unified formulation is proposed to combine the softmax outputs of an ensemble and the groundtruth labels of the training set. The notion “temperature” is first introduced as a parameter to balance the two ingredients. Inspired by [78], we put forward a

recursive learning strategy in Chapter 4, in which knowledge of the attention networks in previous iterations is distilled into the network in the current iteration. During inference, only the current network is stored and evaluated. Compared with directly combining pretrained networks, our proposed recursive learning strategy is able to achieve competitive performance while consuming much less computing resources.

2.4 Filter-based 3D CNNs Pruning

Network pruning techniques can be summarized from two research directions: structured connection pruning and irregular pruning. The first is to remove the entire filters or feature maps to compress the networks, while the latter is to remove any unimportant connections/weights between layers to make the pruned networks sparse. The structured pruning methods can further be sub-categorized as weights-based pruning and activation-based pruning. Subsequently, we introduce the Determinantal Point Processes (DPPs) as optimization method for our proposed pruning goal.

Structured Pruning. Several works have studied removing feature maps or entire filters from a well-trained network. The most prevailing method is weights/filter based pruning. In [79], a cluster-center like method is proposed to select the filters. The concurrent work [80] measures the importance of a filter by the accuracy drop when removing it from the network. [81] proposes the depth multiplier method to scale down the number of filters in each convolutional layer by a factor. Lebedev and Lempitsky [82] use group-sparsity on the convolutional filters to achieve structured brain damage. Wen *et al.* [83] add structured sparsity regularizer on each layer to reduce trivial filters. [84] and [85] prune the redundant filters by adding an regularization term on the whole network to the loss function. [86] learns slim networks by identifying insignificant channels and removing them during training. [87] enforces structured sparsity to the network via introducing additional regularizers. [88] achieves network compression via layer quantization on different levels.

In [89], Ye *et al.* force the outputs of some channels to be constant, and then to prune them by adjusting the biases of their impacting layers.

There are also many activation/feature maps based pruning methods. Polyak and Wolf [90] detect the less frequently activated feature maps with sample input data for face detection applications. Xavier *et al.* [91] apply PCA on feature maps to select the principal ones. Li *et al.* [92] quantify the feature map importance in a feature-agnostic manner to guide model compression. In [93], a subset of feature maps that has the minimum reconstruction error with initial ones is selected. The advantage of structured pruning methods is that they are hardware independent and can be easily applied to standard DCNNs building blocks. But a further retraining process is usually required for reducing the performance loss.

Irregular Connection Pruning. A common approach to reduce the number of parameters is to remove the connections between layers. [94] and [95] propose to remove connections using information drawn from the Hessian of the network’s error function. Sainath *et al.* [96] reduce the number of parameters by analyzing the weight matrices and applying low-rank factorization to the final weight layer. Han *et al.* [97] remove connections with weights smaller than a given threshold. DIVNET [98] seeks diversity among connections and merges similar neurons. Recently, Han *et al.* [99] alternate by pruning near-zero weights, which are encouraged by ℓ_1 or ℓ_2 regularization, and retraining the pruned networks. This line of work preforms connection pruning that leads to irregular network architectures. Thus, these techniques usually require complex sparse representation and delicate hardware customization for practical computational benefits.

Determinantal Point Processes (DPPs). A determinantal point process is a random point process for modelling repulsion. In machine learning, the focus of DPP-based models has been on diverse subset selection from a discrete and finite base set. It was demonstrated in [100] that DPP sampling, which is to select a subset with maximum determinant (det), has superior performance compared with existing subset selection approaches. More importantly, DPP sampling naturally

fulfils our expectations for filter pruning, i.e., the selected filters should be informative and also diverse to each other. Particularly, each point in the distribution is assigned with a magnitude to represent its importance and a direction to measure the similarity to other points. Li *et al.* [101] further propose a fixed size subset selection, k -DPP. As its name implies, k -DPP will select a subset with fixed number of points, e.g., k , with which we are able to set the filter pruning ratios manually so that a trade-off between network compression ratio and performance loss can be achieved.

Chapter 3

Simultaneously Discovering and Localizing Common Objects in Wild Images

3.1 Introduction

Localizing and detecting objects in images are among the most widely studied computer vision problems. They are challenging due to intra-class variation, inter-class diversity, and noisy annotations, especially in wild images. Thus, a large body of fully/strongly annotated data is crucial to train detectors to achieve satisfactory performance. However, manually labeling the presence of objects and even their locations in images is time-consuming, expensive and laborious. On the other hand, the explosive visual data available at almost no cost on websites such as Flickr and Facebook have not been fully utilized. Therefore, building object localization and detection frameworks with weak supervision or even no supervision has been of great interest in recent years.

Weakly-supervised object localization (WSOL) [19, 36] has drawn much attention recently. It aims at localizing common objects across images using the annotations

FIGURE 3.1: Our goal is to simultaneously discover images that contain common objects and also localize common objects in corresponding images. (Clearer when viewed in color.)

TABLE 3.1: Comparison of different supervisions required for the Weakly-/Un-supervised methods and ours.

	Wk.-supervised [36, 102, 103]	Un.-supervised [44–46]	Ours
Image annotations	Yes	No	No
Bounding box annotations	No	No	No
# object categories	One	Many	Many
# noisy images	Not included or only a few	Not included or only a few	Included, as high as 60%

to indicate the presence/absence of the objects of interest. In this chapter, we focus on simultaneously discovering and localizing common objects in real-world images, which shares the same type of output as WSOL, but does not require the annotation of presence/absence of objects. In addition, we tackle this problem in a more challenging scenario where (1) multiple common object classes are contained in the given collection of images, which means this is a totally unsupervised problem; (2) multiple objects or even no object is contained in some of the images (see Figure 3.1), so that there may exist some outliers which do not contain any common objects. It shows in our experiments that even when the outlier ratio reaches as high as 60%, our method can still achieve satisfactory performance. The detailed comparisons of the required supervision for weakly-/un-supervised methods and our proposed method are shown in Table 3.1. It can be clearly seen that, our proposed method could save all annotations on images and bounding boxes and even the number of categories in the given image set, while allowing high rate of noisy images.

FIGURE 3.2: Schematic process of the key working mechanism. Given a set of images, we first extract the object proposals and then these proposals are filtered by a pretrained two-class classifier [7] which can better distinguish the objects from backgrounds. The weighted combination of the proposal’s confidence output from the classifier together with the average saliency within the proposal is the final probability of the proposals containing an object. Then we build a densely connected graph which takes the scored proposals as nodes, and the similarities between these object proposals as the edges. Two models are proposed to handle different scenarios where 1) a given image group contains multiple common object classes, but each image contains at most one target object; 2) an individual image and the image group are both likely to contain multiple common objects. We only show the results under the first assumption in this figure; the common objects are highlighted in green and the outlier images are in red. (Clear when viewed in color.)

3.2 Methods

In this section, we present our proposed approach as shown in Figure 3.2. Given a collection of real world images which may contain outlier/negative images, our goal is to simultaneously discover images that contain common objects and also localize common objects in corresponding images. Moreover, to make the proposed method applicable to wild images, we do not assume to know the total number of object classes in the given dataset, as well as the number of objects that can occur in one single image.

3.2.1 Simultaneous Object Discovery and Localization

To localize the common objects in wild images, we use off-the-shelf object proposals at the first step, then the key problems are 1) to discover the images that contain

at least one common object while eliminating the outlier images; and 2) to locate the proposals corresponding to the common objects. To achieve these two goals simultaneously, we formulate the discovery process as a sub-graph mining problem, where the whole graph is built based on all the object proposals from all images. Once the subset of proposals containing common objects is found, we can easily identify the images that contain these proposals, and the locations of objects in the images are where the corresponding proposals are extracted. In the following, we will introduce how to build graphs in detail based on two different assumptions: 1) there is at most one common object in a single image, which is the most popular setting in previous works and 2) there may be multiple common objects in each image, which is a more challenging scenario.

Given a set of images $\{\mathcal{I}_i\}_{i=1}^N$, and a set of proposals $\{\mathcal{P}_j\}_{j=1}^M$ extracted from all images, we denote an image as positive if it contains the common object; similarly, a proposal is called positive if it corresponds to a common object. Two vectors $\mathbf{g} \in \{0, 1\}^N$, $\mathbf{f} \in \{0, 1\}^M$ are used to infer the labels (positive or negative) of the images and proposals, respectively. As our task is to simultaneously discover and locate common objects, we need to infer $\mathbf{g}$ and $\mathbf{f}$ from the whole image dataset together. It is clear that an image should be labeled positive if one of the proposals extracted from it is positive. The relation between $\mathbf{f}$ and $\mathbf{g}$ can thus be captured by a binary matrix $A \in \mathbb{R}^{M \times N}$, where $A_{j,i} = 1$ if the j th proposal is extracted from the i th image. The correlation of all proposals can be represented by a weighted affinity graph, where each edge represents the similarity between two proposals, and the weight of each node represents the probability of being positive.

Without loss of generality, we first assume that each positive image contains one common object: this scenario is termed as “*Single Instance version of Unsupervised Object Localization (S-UOL)*”. Then the following constraint must hold: $A^\top \mathbf{f} = \mathbf{g}$. This constraint guarantees that there must be one positive proposal ($f_j = 1$) being selected in each positive image ($g_i = 1$), while no positive proposal ($f_j = 0$) will be selected in each negative image ($g_i = 0$). Then finding a sub-graph of positive proposals is to simultaneously infer variables $\mathbf{f}, \mathbf{g}$ which maximize the following

objective:

$$\begin{aligned} \max_{\mathbf{f}, \mathbf{g}} \quad & \mathbf{c}^\top \mathbf{f} - \lambda \mathbf{f}^\top L \mathbf{f} - \eta \|\mathbf{g}\|_0 \\ \text{s.t.} \quad & A^\top \mathbf{f} = \mathbf{g} \end{aligned} \quad (3.1)$$

where each element c_j in $\mathbf{c} \in \mathbb{R}^M$ is the confidence of the j th proposal being positive. The first term $\mathbf{c}^\top \mathbf{f}$ aims at maximizing the cumulative score of the selected sub-graph. Laplacian matrix L is defined as $L = D - W$, where $W \in \mathbb{R}^{M \times M}$ is the similarity matrix based on the proposal features, and D is the diagonal matrix in which $D_{j,j} = \sum_{k=1}^M W_{j,k}$. The second term $\mathbf{f}^\top L \mathbf{f}$ is to minimize the effect of negative proposals by emphasizing more on the edge connections with higher weights, and the parameter λ controls the influence of this connectivity. The last term $\|\mathbf{g}\|_0$ attempts to eliminate negative images. The reason for combining the outlier discovery and object localization into a joint formulation is that the two processes can help each other during learning, *e.g.*, the more frequently occurring images are more likely to contain common objects, and vice versa. It is worth noting that most previous studies [44, 45] only consider image sets without outliers or with a small proportion of outliers, however, our method can handle data with a large portion of outlier images.

The above formulation can be easily extended to address a more general scenario where for a group of images, some of them may contain multiple common objects. We call this setting as “*Multiple Instance version of Unsupervised Object Localization (M-UOL)*”. Then the objective can be written as:

$$\max_{\mathbf{f}} \mathbf{c}^\top \mathbf{f} - \lambda \mathbf{f}^\top L \mathbf{f} - \eta \|\mathbf{f}\|_0 \quad (3.2)$$

Definitions of variables are the same as Equation (3.1), but we do away the constraint as it is unknown how many positive proposals each positive image may contain. Note that we reward sparse solutions by the last term due to that there are dozens of proposals extracted from each image and the majority of them tend to be negative ones. For this reason, a sparsity regularizer on $\mathbf{f}$ is still necessary

FIGURE 3.3: Illustration of sub-graph mining from a densely connected graph. The nodes of the graph are represented by circles in different sizes to indicate the confidence of proposals containing common objects. The different numbers in circles denote the proposals are extracted from different images. The edges of the graph are represented by the distances between circles, the more similar two proposals are, the closer they are in the figure. Each node is connected to the nearest 5 neighbors. The resulting sub-set of nodes are highlighted by green which enjoying high confidences and are cohesive with each other. Although close to each other, the red nodes are unlikely to contain object due to low weights of these nodes, and some of the grey ones in spite of enjoy high confidences but are unlikely to be common objects since they are unique to other nodes. (Viewed in color with magnification.)

for co-localization problems to avoid selecting large numbers of proposals. The parameter η controls the influence of sparsity. The *M-UOL* model can also be applied to the problem of image and instance retrieval, where the node confidences represent how relevant the reference images and the query are, and the similarity matrix W is composed of the similarity between reference images. Different from traditional retrieval methods which only consider the relation between query and reference images, our method can also take advantage of the relations among the reference images.

Compared with most of the previous graph-based methods, we make a significant improvement by leveraging both node and edge weights to build graphs. The roles of the nodes and edges on the sub-graph mining results are shown in Figure 3.3. Based on our formulation, the isolated nodes even with high possibilities of containing an object are less likely selected, such as the node corresponding to a bike (not

a common object) in Figure 3.3. A set of cohesively connected nodes may also not be the ideal sub-graph if the weight of each node is low (*i.e.*, less likely to be objects), such as the cluster of sky highlighted in red in Figure 3.3. Only nodes enjoying high confidence of being the objects and correlating cohesively with each other are selected (*e.g.*, patterns of airplanes and cats), which are exactly the first and second terms in Equation (3.1) and Equation (3.2) designed for.

3.2.2 Optimization

To solve the sub-graph mining problems defined in Equation (3.1) and Equation (3.2), we resort to the maximal flow algorithm proposed by Boykov and Kolmogorov [104]. It has been demonstrated in [105, 106] that a minimum cut can be efficiently computed with the maximum flow algorithm; meanwhile, it can be proved that our objective functions are combinations of cut-functions, so that to optimize our objective functions is equivalent to find a min-cut on a graph.

Proof. Given a graph $\mathcal{G}$ over nodes $V = \{1, \dots, M\}$, the goal of minimum cut is to find a sub-set $S \subset V$ that minimizes the cut-function $\sum_{p \in S} \sum_{q \notin S} E_{p,q} = \sum_{p=1}^M \sum_{q=1}^M f_p(1 - f_q)E_{p,q}$, where E denotes the adjacency matrix of the graph, f_p is 1 if $p \in S$ and 0 otherwise. The graph-regularized subset selection problem formalized in Equation (3.1) can be represented as following by substituting the constraint into the objective function:

$$\min_{\mathbf{f} \in \{0,1\}^M} (\eta \|A^\top \mathbf{f}\|_0 - \mathbf{c}^\top \mathbf{f}) + \lambda \mathbf{f}^\top L \mathbf{f} \quad (3.3)$$

By introducing artificial nodes s and t to the graph, the first term can be encoded as a *cut-function*:

$$\begin{aligned}
& \eta \|A^\top \mathbf{f}\|_0 - \mathbf{c}^\top \mathbf{f} \\
&= \eta \sum_{i=1}^N \sum_{p=1}^M A_{p,i} f_p - \sum_{p=1}^M c_p f_p \\
&= \eta \sum_{p=1}^M f_p \sum_{i=1}^N A_{p,i} - \sum_{p=1}^M c_p f_p \\
&= \eta \sum_{p=1}^M f_p - \sum_{p=1}^M c_p f_p \\
&= \sum_{p=1}^M (\eta - c_p) f_p \\
&= \sum_{p \in S, c_p < \eta} (\eta - c_p) + \sum_{p \in V, c_p \geq \eta} (\eta - c_p) - \sum_{p \notin S, c_p \geq \eta} (\eta - c_p) \\
&= \sum_{p=1}^M E_{s,p} f_s (1 - f_p) + \sum_{p=1}^M E_{p,t} f_p (1 - f_t) + \mathcal{C}
\end{aligned} \tag{3.4}$$

where $\mathcal{C} = \sum_{p \in V, c_p \geq \eta} (\eta - c_p)$ is a constant, $f_s = 1, f_t = 0$ and the edges of the two augmented nodes s and t are given by $E_{p,q} = \lambda W_{p,q}$ for $1 \leq p, q \leq n$ and $E_{s,p} = \max\{c_p - \eta, 0\}$ and $E_{t,p} = \eta - c_p + E_{s,p}$, W is the adjacency matrix of the original graph without augmentation. As $f_s = 1$ and $f_t = 0$ enforce that $s \in S$ and $t \notin S$, it follows that Equation (3.1) is an s/t min-cut problem on the transformed graph defined by the adjacency matrix E over the nodes of $\mathcal{G}$ augmented by s and t . The aforementioned situation still holds if W is a weighted adjacency matrix, so the min-cut reformulation can also be applied to a weighted network.

The second term of Equation (3.3) can be easily represented as a cut-function over graph $\mathcal{G}$:

$$\begin{aligned}
\mathbf{f}^\top L \mathbf{f} &= \sum_{p=1}^M f_p \left(D_{p,p} - \sum_{q=1}^M W_{p,q} f_q \right) \\
&= \sum_{p=1}^M \sum_{q=1}^M W_{p,q} f_p (1 - f_q)
\end{aligned} \tag{3.5}$$

Similarly, it can be proved that the objective Equation (3.2) can also be represented as the combination of two cut-functions. $\square$

It is clearly shown from the above proof that both of the proposed objectives could be represented by cut-functions, thus could be optimized by maximal flow algorithms. In our implementation, we use Boykov-Kolmogorov algorithm [104] to solve our node selection problems. Although the worst case complexity is in $\mathcal{O}(M^2en)$, where e represents the number of edges in the graph and n is the size of the minimum cut, it performs efficiently in practice since the graph is rather sparse. Comparing our *S-UOL* Model with the joint model proposed in [44], one can find that our method can directly output the binary solution while [44] needs to set an extra threshold for the binarization of their real-value solution.

3.2.3 Ablation Simulation

To visualize the effectiveness of our proposed framework and its components, we run a test experiment on some simulated 2D data points (see Figure 3.4) based on the general assumption that there may exist multiple common objects in a single positive image. In this simulation experiment, the node confidence is set as its density calculated following [107], and the similarity matrix is calculated by $W_{k,l} = \exp(-\|\mathbf{x}_{i,k} - \mathbf{x}_{j,l}\|^2)$. The affinity graph is constructed in the same way as described in Section 3.2.1. The proposed sub-graph mining technique is tested on these data, and the results are shown in Figure 3.4. Each point represents a node, and the green ones are positive, while red ones are negative. There are 1600 positive points and 2000 negative points in total. In order to show the relative importance of each part in the affinity graph formulated in Equation (3.2), we display the results, which are optimized based on the nodes only (Unary term in Equation (3.2)), on the edges only, and on both the nodes and edges (Unary and Laplacian term in Equation (3.2)), in the first row. Figure 3.4(b) pictures the case where we treat the top 2000 points with higher node confidences as positive. It can be seen that some false positives will be chosen if we only rely on the node

FIGURE 3.4: The simulation results using *M-UOL* Model. **Top:** Groundtruth, and results using only node confidence, edge similarity and both. **Middle:** Results vary in terms of different values of η with the optimal λ (grid-search on ranges of values). **Bottom:** Results vary in terms of different values of λ with η being fixed as 0.3. (Better viewed in color.)

scores. Figure 3.4(c) shows results of the 1600-nearest-neighbor degree of the nodes, although the majority of the positive points are choose, the rate of false negative is rather high. From Figure 3.4(d), which shows the case depending on both nodes and edges but without the regularization term, we can find that almost all the positive points are correctly recalled, but the false positive rate is so high, thus a regularization term is necessary for removing those false positive points.

The results on the joint objective (Equation (3.2)) are visualized in the second and third row, with various η and λ showing the influence of the two parameters on the results. Compared with Figure 3.4(d), it can be found that with the regularization term, most of the false positive points can be eliminated, and the coefficient η controls how sparsity of the results $\mathbf{f}$ could be. The parameter λ influences the

results by tuning the relative importance of the node confidences and the similarities between nodes. We should note that there is no standard metric for common object or common pattern discovery since the definition of “*common*” depends on the dataset and the task, so theoretically there are no optimal parameters of η and λ . In the following experiments on the real-world images, the parameters are fixed to fulfill the evaluation metric of the specified task.

3.2.4 Applications to Real-world Images

To utilize the proposed framework for common object discovery and localization in real-world images, we first extract object proposals for each image using off-the-shelf method [16]. After generating a pool of object proposals, we then resort to a pre-trained two-class classifier, called DeepBox [7], to distinguish the objects from backgrounds. The proposals with lower probabilities of containing objects are removed, so that the size of proposal pool is much smaller.

As CNN [108] has been demonstrated to have the capability to capture high-level image representations, we thus employ the popular AlexNet [6] to extract features for region proposals. The AlexNet is finely trained on the ImageNet dataset with 1000 outputs at the last layer, we use the outputs of the second fully-connected layer, that is 4096-dimensional feature for each proposal.

Proposal Confidence. We introduce a prior for each proposal to represent the confidence of being positive. The average saliency [109] within the box, which provides useful common foreground prior and has been extensively used in foreground co-segmentation and image co-localization [44, 46], can be used as an instructor of positive proposals. Furthermore, the scores of DeepBox [7], which is used at previous step to filter out proposals based on a binary classifier, could also be used to indicate the objectness of proposals. Finally, our proposal score $\mathbf{c}$ in the first term of our objective functions can be expressed as the weighted combination of the above two pieces of prior information, the weight is fixed as 0.7/0.3 in our

experiments. It is noted that the choice of proposal scores can be so flexible and it depends on the specific task and the prior knowledge.

Similarity Matrix. Given the proposal representations, we compute the similarity matrix $W_{i,j}$ with all the proposals. Based on the observation that the number of positive proposals accounts for only a small proportion of the total proposals and the similarity matrix should be sparse, we set the similarities between proposals from the same image to be 0 for the proposed “*S-UOL*” Model. On the other hand, for the “*M-UOL*” Model where positive images containing multiple common objects, we set the similarities to be 0 only when the IoU of two proposals is larger than a certain value (*e.g.*, $\frac{\text{area}(\mathcal{P}_i \cap \mathcal{P}_j)}{\text{area}(\mathcal{P}_i \cup \mathcal{P}_j)} > 0.5$). In addition, the similarity W_{ij} is also set to be 0 if the i th and j th proposals are not the K -reciprocal neighbors to each other, K is fixed as the number of images in our experiments. This makes sure the quality of the edge connections. We summarize the working flow of our proposed unsupervised common object discovery framework in Algorithm 1.

Algorithm 1 Cohesive sub-graph mining

Input: A set of unlabeled images $\{\mathcal{I}_i\}_{i=1}^N$.

Output: The proposal labels f and image labels g if necessary.

- 1: Extract object proposals using any off-the-self methods, denoted by $\{\mathcal{P}_j\}_{j=1}^M$ and the corresponding index matrix $A \in \mathbb{R}^{M \times N}$;
 - 2: Compute the confidence $\{c_j\}_{j=1}^M$ of being positive for each proposal based on prior knowledge;
 - 3: Extract features from each proposal, and compute the similarity matrix and Laplacian matrix, denoted respectively by $W, L \in \mathbb{R}^{M \times M}$;
 - 4: Formulate the task as *S-UOL* Model or *M-UOL* Model according to the assumption whether the positive images only contain a single common object;
 - 5: Solve the *S/M-UOL* Model using our proposed optimization method to get the proposal labels $f \in \mathbb{R}^M$ (and image labels $g \in \mathbb{R}^N$ if necessary).
-

3.3 Unsupervised Object Discovery Experiments

To evaluate the performance of our proposed method on co-localization and discovery, we conduct experiments on two benchmarks: the Object Discovery dataset [22] and the PASCAL VOC 2007 dataset [110]. The proposed method is compared with

some representative weakly-supervised methods [19, 36, 102, 103, 111–114] and also fully unsupervised methods [22, 44–46, 48, 115–117] in terms of the detected locations of common objects. To evaluate the robustness of our method in the presence of outlier images, we also conduct experiments on the Object Discovery dataset with outlier ratios ranging from 0% to 60%.

Following [45], we implement two types of settings: 1) The *separate-class* experiment, which is conducted across images with the same class labels (for positive images), so that there is only one common object class in such a setting. It should be noted that the supervision of this setting is even weaker than traditional weakly-supervised setting where all images are labeled; however, there can exist anonymous noisy images which do not contain any common objects in our setting, and the number of noisy images is unknown either. 2) The *mixed-class* experiment, which is in a fully unsupervised way without any prior knowledge and there may be multiple common object classes contained in the given image sets. This scenario is common for image/object search, object discovery in videos, etc. In such a case, the positive images may contain different types of objects, while the negative images are unknown. This problem has not been well studied before. To save space, we abbreviate *separate-class* to “sep.” and *mixed-class* to “mix.”, and we use “sep.-c” to denote the setting where only the nodes are considered (removing the Laplacian term in the objective) during mining the subgraphs.

Implementation Details: We first extract 300 proposals from each image using [16]. We then remove the proposals predicted as backgrounds by [7]. The number of the remaining object proposals varies from 32 to 180. Grid-search experiments over ranges of values are conducted to find the parameters in the two proposed models, *i.e.*, *S-UOL* and *M-UOL*. For the *S-UOL* Model, the correct localization (CorLoc) metric is adopted for a fair comparison with previous methods, which is defined as the percentage of images correctly localized according to the PASCAL criterion:

$$\frac{\text{area}(\mathcal{P}_i \cap \mathcal{P}_{gt})}{\text{area}(\mathcal{P}_i \cup \mathcal{P}_{gt})} > 0.5,$$

where $\mathcal{P}_i$ is the predicted box and $\mathcal{P}_{gt}$ is the ground-truth box. For these images which contain multiple objects, we follow the commonly used criterion that regards the image as correctly co-localized if any of its objects are localized correctly.

For the *M-UOL* Model, since the last term in the objective function (*i.e.*, Equation (3.2)) only controls the number of selected object proposals in total and there is no supervision or assumption to indicate the exact number of common objects in each image. It is likely that the resulting number of proposals in each image may be more or less than the number of ground truth objects. Thus in our experiments, the *M-UOL* Model is evaluated by F1-score which is a measurement in instance-level rather than the image-level metric CorLoc adopted in the *S-UOL* Model. The F1-score is defined as:

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

where precision is the proportion of correctly localized positives (IoU>0.5) relative to the number of identified positives, and recall is the proportion of correctly localized objects relative to the total number of ground truth objects.

3.3.1 The Object Discovery dataset

The Object Discovery dataset [22] contains 15k images in three categories: airplane, car, and horse. As it is automatically collected by the Bing API using image queries of these three categories, the dataset contains outlier/negative images without the three common object classes. To compare with previous co-segmentation and co-localization method fairly, we use a subset of the dataset containing 100 images for each category. There are 18, 11, 7 outliers for three categories, respectively. The dataset is originally designed for co-segmentation with pixel-level labels. We convert the ground-truth segmentations to object locations by drawing tight bounding boxes around positive pixels.

TABLE 3.2: CorLoc(%) for the Object Discovery subset using *S-UOL* Model.

Method	Airplane	Car	Horse	Av.
Kim <i>et al.</i> [118]	21.95	0.00	16.13	12.69
Joulin <i>et al.</i> [48]	32.93	66.29	54.84	51.35
Joulin <i>et al.</i> [115]	57.32	64.04	52.69	58.02
Rubinstein <i>et al.</i> [22]	74.39	87.64	63.44	75.16
Tang <i>et al.</i> [44]	71.95	93.26	64.52	76.58
Cho <i>et al.</i> [45]-sep.	82.93	94.38	75.27	84.19
Cho <i>et al.</i> [45]-mix.	81.71	94.38	70.97	82.35
Ours-sep.-c	69.73	89.0	68.19	75.64
Ours-sep.	82.1	95.26	76.0	84.45
Ours-mix.	81.13	94.17	72.52	82.61

Table 3.2 shows the comparison of state-of-the-art methods and our *S-UOL* Model, which assumes that each image contains at most one common object, in the separate-class and mixed-class settings. From the table we can see that based on the separate-class setting (Ours-sep.), our method is comparable to the state-of-the-art [45]. The gap between “sep.-c” and “sep.” demonstrates that both nodes and edges can contribute to the final performance. Although the results on the mixed-class setting are about 2% lower on average compared with the separated-setting, our method still outperforms the majority of previous methods. The reason for the gap between results of these two settings is that the separate-class experiments are conducted in each category with 100 images so that there is only one common object class in this case and the mixed-class experiment is based on all the three categories with 36 negative images in total.

In Figure 3.5, we show the performance of CorLoc over outlier ratios using the proposed *S-UOL* Model. The outlier images are randomly sampled from PASCAL VOC 2007 (except the images labeled airplane, car and horse). In the separate-class experiment where there is only one common object class among the image sets, the performance is still satisfactory even when the outlier ratio reaches as high as 60%. In the mixed-class experiment where there are three common object classes, the performance is steady at lower outlier ratios, but drops heavily at high outlier

FIGURE 3.5: CorLoc(%) for the Object Discovery dataset with outliers using our *S-UOL* Model.

TABLE 3.3: F1-scores for the Object Discovery subset using the *M-UOL* Model.

Method	Airplane	Car	Horse	Av.
EdgeBox [16]	9.99	11.79	8.48	10.09
DeepBox [7]	48.59	60.9	39.2	49.56
Ours-sep.	83.07	82.03	76.81	80.64
Ours-mix.	71.14	72.44	64.07	69.22

ratios. This is because, with high outlier ratio, such as 50%, each of the common object class only accounts for about 16% which is already not so “dominant”.

We compare the proposed *M-UOL* Model with EdgeBox [16] and DeepBox [7] in terms of the F1-score. To evaluate the performance, we use the top 300 object proposals for the EdgeBox and the positives identified by DeepBox from the 300 proposals are used to evaluate itself. The results are listed in Table 3.3.

Figure 3.6 shows example results using the proposed *S-UOL* Model and *M-UOL* Model in the mixed-class setting. In spite of accounting for small size, the *S-UOL* is able to localize the target objects from the cluttered backgrounds. It is noted that for the *M-UOL* Model, we have no constraint on the exact number of objects in each image. Thus, the output proposals are usually more or less than the ground truth objects. We also show some outlier images that can be successfully eliminated by both models.

FIGURE 3.6: Examples of successful co-localization results for the Object Discovery subset in mixed-class setting. **Left:** Results using our *S-UOL* Model; **Right:** Outlier/Negative images.

3.3.2 PASCAL VOC 2007 dataset

The PASCAL VOC 2007 dataset [110] is one of the most popular datasets for computer vision tasks such as classification, detection, and segmentation, and it contains 5011 images in 20 object categories. This dataset is quite challenging for the task of co-localization because: 1) most of the objects in this set of images are with considerable clutter, occlusion, and different viewpoints; 2) many images in this dataset contain multiple common objects. Following the experimental setup defined in [45], we evaluate our method on two sizes of sets: PASCA07-6 $\times$ 2 and PASCA07-all. The PASCA07-6 $\times$ 2 consists of all images from 6 classes (aeroplane, bicycle, boat, bus, horse, and motorbike) of train+val dataset from the left and right aspects each. Each of the 12 class/viewpoints combinations contains between 21~50 images for a total of 463 images. And for the large-scale dataset PASCA07-all, we use all images from train+val datasets to evaluate our method.

PASCA07-6 $\times$ 2. In Table 3.4, we compare our *S-UOL* model with previous methods in terms of CorLoc. In both separate-/mixed-class settings, our method can

TABLE 3.4: CorLoc(%) for PASCAL07-6 $\times$ 2 using the *S-UOL* Model. The methods labeled by ‘*’ are in weakly supervised way.

Method	aeroplane		bicycle		boat		bus		horse		motorbike		Av.
	L	R	L	R	L	R	L	R	L	R	L	R	
*Siva and Xiang <i>et al.</i> [119]	-	-	-	-	-	-	-	-	-	-	-	-	49
*Pandey <i>et al.</i> [113]	-	-	-	-	-	-	-	-	-	-	-	-	61.05
Tang <i>et al.</i> [44]	41.86	51.28	25.00	24.00	11.36	11.63	38.10	56.52	43.75	52.17	51.28	64.71	39.31
*Cho <i>et al.</i> [45]-sep.	62.79	71.79	77.08	62.00	25.00	32.56	66.67	91.30	83.33	86.96	82.96	70.59	67.68
Cho <i>et al.</i> [45]-mix.	62.79	66.67	54.17	56.00	18.18	18.60	42.86	69.57	70.83	71.74	69.23	44.12	53.73
Niu <i>et al.</i> [117]	64.28	87.17	35.41	39.13	16.66	30.00	42.86	60.86	46.80	53.33	59.45	75.75	50.97
*Ours-sep.-c	49.23	62.46	39.0	32.07	37.64	34.58	42.57	68.42	54.83	62.46	58.0	70.17	50.94
*Ours-sep.	67.44	76.92	52.08	40.0	45.45	48.84	57.14	95.65	79.17	84.78	74.36	85.29	67.26
Ours-mix.	62.79	71.79	41.67	34.0	24.91	39.53	47.62	81.96	64.58	71.74	69.23	64.41	56.19

FIGURE 3.7: Examples of co-localization for the PASCAL07-6 $\times$ 2 using the *S-UOL* Model under mixed-class setting.TABLE 3.5: Average CorLoc(%) for PASCAL07-6 $\times$ 2 using the *S-UOL* Model.

Method	Average Corloc
Deselaers <i>et al.</i> [120]	50
Siva and Xiang <i>et al.</i> [119]	49
Tang <i>et al.</i> [44]	39
Cho <i>et al.</i> [45]-sep.	68
Cho <i>et al.</i> [45]-mix.	54
Ours-sep.	67
Ours-mix.	56

achieve satisfactory performance. The results of the separate-class without the Laplacian term (sep.-c) is also among the top performances. Though being lower in certain categories, the average performance of our method is comparable to that of [45] in the separate-class setting (sep.) and much higher in the mixed-class setting. Figure 3.7 presents some examples generated by the *S-UOL* Model. And the results of the *M-UOL* Model for this dataset are showed in Table 3.7 with the average F1-scores.

PASCA07-all. Here we tackle a much more challenging and large-scale discovery

TABLE 3.6: CorLoc(%) for PASCAL07-all using the *S-UOL* Model. The methods labeled by ‘*’ are in weakly supervised way.

Method	aero	bicy	bird	boa	bot	bus	car	cat	cha	cow	dtab	dog	hors	mbik	pers	pint	she	sofa	traf	tv	Av.
*Pandey <i>et al.</i> [113]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	30.31
*Shi <i>et al.</i> [114]	54.7	22.6	33.7	24.5	4.6	33.9	42.5	57.0	7.3	39.1	24.1	43.3	41.3	51.5	25.3	13.3	28.0	29.5	54.6	11.8	32.1
*Shi <i>et al.</i> [36]	67.3	54.4	34.3	17.8	1.3	46.6	60.7	68.9	2.5	32.4	16.2	58.9	51.5	64.6	18.2	3.1	20.9	34.7	63.4	5.9	36.2
*Cinbis <i>et al.</i> [19]	56.6	58.3	28.4	20.7	6.8	54.9	69.1	20.8	9.2	50.5	10.2	29.0	58.0	64.9	36.7	18.7	56.5	13.2	54.9	59.4	38.8
*Li <i>et al.</i> [102]	73.1	45.0	43.4	27.7	6.8	53.3	58.3	45.0	6.2	48.0	14.3	47.3	69.4	66.8	24.3	12.8	51.5	25.5	65.2	16.8	40.0
*Li <i>et al.</i> [103]	64.3	54.3	42.7	22.7	34.4	58.1	74.3	36.2	24.3	50.4	11.0	29.2	50.5	66.1	11.3	42.9	39.6	18.3	54.0	39.8	41.2
*Bilen <i>et al.</i> [121]	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
*Kantorov <i>et al.</i> [122]	78.8	66.7	52.9	25.0	26.3	68.0	73.6	44.8	14.9	62.3	45.2	46.3	61.6	82.3	35.3	39.6	69.1	30.9	62.0	69.5	52.8
*Wang <i>et al.</i> [112]	80.1	63.9	51.5	14.9	21.0	55.7	74.2	43.5	26.2	53.4	16.3	56.7	58.3	69.5	14.1	38.3	58.8	47.2	49.1	60.9	48.5
*Jie <i>et al.</i> [123]	70.2	60.0	53.9	26.1	28.3	58.9	75.4	58.9	14.8	63.4	17.9	52.6	51.7	67.0	19.7	46.3	63.9	42.4	67.0	65.1	50.2
Joulin <i>et al.</i> [46]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	24.6
*Cho <i>et al.</i> [45]-sep.	50.3	42.8	30.0	18.5	4.0	62.3	64.5	42.5	8.6	49.0	12.2	44.0	64.1	57.2	15.3	9.4	30.9	34.0	61.6	31.5	36.6
Cho <i>et al.</i> [45]-mix.	40.4	32.8	28.8	22.7	2.8	48.4	58.7	41.0	9.8	32.0	10.2	41.9	51.9	43.3	13.0	10.6	32.4	30.2	52.7	21.8	31.3
*Ours-sep-c	31.8	25.2	33.5	22.0	16.2	38.3	40.3	38.6	13.6	24.6	10.9	26.0	31.1	38.3	28.5	19.7	50.0	20.7	40.1	11.4	29.4
*Ours-sep.	55.2	35.7	45.9	30.0	20.2	49.2	50.2	53.2	18.1	44.1	13.6	49.1	39.7	43.3	35.4	22.3	67.7	29.7	50.5	25.3	38.9
Ours-mix.	42.2	31.7	36.7	20.0	16.2	36.2	40.2	51.9	15.1	40.0	12.6	44.6	31.5	38.9	31.3	19.7	54.8	28.7	43.9	8.9	32.2

TABLE 3.8: Comparison of different ways of selecting proposals.

Selected pps.	Object Discovery	VOC07-6×2	VOC07-all
All	82.97	56.48	32.19
Top-50%	81.29	55.74	31.08
Top-30%	80.95	54.58	30.42
DeepBox [7]	82.61	56.19	32.21

TABLE 3.9: Average CorLoc(%) in terms of various weights using S-UOL Model.

saliency / objectiveness	Object Discovery		VOC07-6×2		VOC07-all	
	Ours-sep.-c	Ours-mix.	Ours-sep.-c	Ours-mix.	Ours-sep.-c	Ours-mix.
0.9 / 0.1	76.18	81.25	48.39	55.16	28.01	30.44
0.7 / 0.3	75.64	82.61	50.94	56.19	29.42	32.19
0.5 / 0.5	72.48	79.82	49.77	55.97	29.31	33.75
0.3 / 0.7	70.76	75.94	47.89	53.31	27.99	30.73

TABLE 3.10: Comparison of CNN-based features with hand-crafted features.

Feature	Object Discovery	VOC07-6×2	VOC07-all
CNN	82.61	56.19	32.21
SIFT	81.23	55.92	31.47
HOG	82.05	54.87	30.73

[122]. Table 3.7 shows the comparisons of our *M-UOL* Model with EdgeBox [16] and DeepBox [7]. We visualize some experimental results in Figure 3.8.

3.3.3 Discussion

In this section, we conduct more ablation studies to evaluate (1) different strategies of proposal selection; (2) various weighted combinations of the saliency confidence and objectiveness; (3) the CNN-based features and traditional hand-crafted features.

Table 3.8 shows the average CorLoc(%) on object discovery benchmarks. “All” denotes using all 300 proposals extracted by [16], and “Top- $x\%$ ” denotes using top $x\%$ proposals ranked by objectness score [16]. The average number of selected proposals by DeepBox [7] is 67 for each image. From the table we can see that, using all proposals is slightly better than Deepbox. Actually, the main advantage of using DeepBox is to reduce computational complexity. As we stated in Section 3.2.2, the

worst case complexity of optimizing objective function is $\mathcal{O}(M^2en)$, where M is the number of proposals from all images. With DeepBox, the average number of selected proposals can be reduced to 67 per image, which is significantly more efficient compared to 300 per image. To further improve the proposal confidence, we combine the scores from DeepBox and the average saliency. Table 3.9 shows the performance changes with different combination weights. We use the 0.7/0.3 in our experiments as it achieves the most promising performance in average.

To validate our method in fully-unsupervised way, we replace the CNN features with HOG and SIFT features to calculate the similarity matrix while keeping other implementation details unchanged. The results (Cor-Loc(%)) based on the *S-UOL* Model in the mixed-class setting are shown in Table 3.10. Even with HOG or SIFT features, our method can still obtain good results.

3.4 Image Retrieval Experiments

Our proposed framework can be easily applied to the problem of image/instance retrieval. Different from traditional image/instance retrieval methods, which only depend on the similarity between query and reference images to rank the references, we also use the similarities among references. Specifically, the initial retrieval score which is obtained by similarities between query and references is used as node confidence and the similarities among references are used to construct the Laplacian matrix in our objective function. The idea to apply our method to improve ranking is based on the following observation. If the neighbor images of an image I match query I_q , then it is likely that image I also matches I_q , even with a mild similarity between I and I_q . On the contrary, if the neighbor images of an image I are distinct to the query I_q , then it is likely that I do not match I_q either, even though the individual similarity between I and I_q is high, so that noisy images can be removed. The output of our proposed algorithm is a binary vector with 1 denoting an image belongs to a specified query, then the selected images are re-ranked by the initial retrieval score. We evaluate our proposed framework on two datasets, Oxford5k

building [124] and Holidays dataset [125]. The first is selected for instance/object retrieval, and the latter for scene/image retrieval. Figure 3.9 shows some samples of the two datasets. For both datasets, we report mAP as the measurement metric. To save space, we use ‘-c’ to represent the case that only considering the similarity between query and reference images.

FIGURE 3.9: Samples of the two datasets, one for image retrieval and another for instance retrieval.

3.4.1 Holidays

This dataset contains 1491 images of which 500 are queries. It contains images of different scenes, items and monuments. For each image, the maximum activation of convolutions (MAC) feature and regional maximum activation of convolutions (R-MAC) feature [51] are extracted. The final dimensions of the two kinds of features are both 512. Let $\mathbf{q} \in \mathbb{R}^d$ denote the feature vector of a query, and $I \in \mathbb{R}^{d \times N}$ be the feature matrix of the dataset, where N is the number of reference images. Then the node confidence $\mathbf{c} \in \mathbb{R}^N$, which represents the similarity between the query and the reference images, can be computed as:

$$\mathbf{c} = \mathbf{q}^\top I, \quad (3.6)$$

TABLE 3.11: The results of image retrieval on Holidays dataset.

Method	Dimension	mAP
Sharif <i>et al.</i> [49]	500	82.2
Perronnin <i>et al.</i> [126]	512	82.7
	4k	84.7
Arandjelovic <i>et al.</i> [50]	256	79.9
Radenović <i>et al.</i> [127]	512	82.5
Rezende <i>et al.</i> [128]	512	82.3
R-MAC: Ours-c	512	84.5
R-MAC: Ours	512	87.1
MAC: Ours-c	512	83.2
MAC: Ours	512	85.6

and the similarity matrix $W \in \mathbb{R}^{N \times N}$ between the reference images are computed as:

$$W = I^\top I. \quad (3.7)$$

Table 3.11 lists our performance and some recent studies on this dataset. We can find that with similar dimension of feature vector, our proposed *M-UOL* model can improve $> 2\%$ compared with the results using only the similarity between query and reference images. Note that any state-of-the-art methods which learn the similarity between the query and reference images can be used as the node confidences in our framework, then the retrieval accuracy could be further improved by our proposed *M-UOL* model.

3.4.2 Oxford5k

This dataset is composed of 5063 reference images and 55 queries. The object bounding boxes in the queries have been given, so it means the query objects are known. The challenging of this dataset is that the scale of the objects in the reference images varies a lot, and all the buildings are very similar. Since this dataset targets at instance retrieval, the objects contained in images should be

TABLE 3.12: The results of instance retrieval on Oxford5k.

Method	Dimension	mAP
Babenko <i>et al.</i> [129]	256	65.7
Tolias <i>et al.</i> [51]	512	66.9
	-	77.3
Arandjelovic <i>et al.</i> [50]	256	63.5
Radenović <i>et al.</i> [127]	512	77.0/80.1
Rezende <i>et al.</i> [128]	512	64.1
MAC <i>et al.</i> [33]	512	73.9
R-MAC: Ours-c	512	77.6
R-MAC: Ours	512	80.2
MAC: Ours-c	512	73.9
MAC: Ours	512	76.3

focused, thus we extract object proposals from each image. Then for each object proposal, the maximum activation of convolutions (MAC) feature and regional maximum activation of convolutions (R-MAC) feature [51] are extracted. The final dimensions of the two kinds of features are both 512. We denote the feature matrix of all the object proposals as $P \in \mathbb{R}^{d \times M}$. In such a case, the node confidence is $\mathbf{c}' \in \mathbb{R}^M$, which represents the similarity between the query object and the object proposals:

$$\mathbf{c}' = q^\top P, \quad (3.8)$$

and the similarity matrix $W' \in \mathbb{R}^{M \times M}$ between the object proposals is computed as:

$$W' = P^\top P. \quad (3.9)$$

The node confidence of each reference image, *i.e.*, the distance between query object to the reference image can be determined by the nearest object proposals in the image. The similarity W_{ij} between any two images i and j is determined by averaging the similarities between all the object proposals in them. From Table 3.12, we can see that with similar feature dimension, our *M-UOL* model using R-MAC can achieve state-of-the-art performance. Compared with the retrieval results of using

only the similarity between the query and reference images (“Ours-c”), our *M-UOL* model can increase about 3% with both MAC and R-MAC, which demonstrates the effectiveness of our proposed *M-UOL* model.

We select the retrieved images from instance search in Oxford5k dataset to show the failure cases. In Figure 3.10, each row shows a query and the retrieved images based on it, and the incorrect returns are highlighted by red. Note that for instance retrieval task, the goal is to find images containing patches that are similar to the instances, instead of returning similar images. For the second query, our method fails on the image of the second column because it is similar to the other correctly returned images. For the third query, both of the incorrect returns are similar to the given query and also the other returns. This case is difficult to both conventional methods and our proposed method as the similarity between query and reference image is critical for image retrieval task. The possible reasons for the failure cases in the last row is that the appearance similarity and outline similarity are both important for the instance search.

FIGURE 3.10: Examples of instance retrieval results on Oxford5k.

3.5 Conclusion

In this chapter, we propose a framework for common object discovery and localization in wild images. Like most previous methods which are based on the assumption that there is only one object contained in each positive image, we introduce the *S-UOL* Model, which has been demonstrated to be robust even with a significant proportion of outlier images. Then, the *M-UOL* Model is further introduced for a more general scenario where there could be multiple common objects contained in each image. The proposed *M-UOL* Model can be easily extended to the task of image/instance retrieval. Inspired by min-cut/max-flow algorithms, we then present a constrained sub-graph mining algorithm to optimize the two models. To evaluate the proposed method, we conduct extensive experiments under multiple settings and compare with many representative studies. Empirical results demonstrate that our proposed method performs well despite in a fully unsupervised way, and even when variations in scale, view-point, appearance or partial occlusions frequently occur in the images.

Chapter 4

Attention-Aware Noisy Label Learning for Image Classification

4.1 Introduction

In recent years, deep convolutional neural nets (CNNs) have triumphed over many vision tasks [130–132]. However, the overwhelming performances of CNNs heavily rely on large-scale high-quality labeled data, e.g., ImageNet [133], which are typically laborious and costly to collect and annotate. Nevertheless, there are millions of freely available images with user-supplied labels that can be easily accessed from the web. Although utilizing web images has become a popular research direction in the field of multimedia processing, the performance is inferior to its counterpart on finely labeled data. The main reason is that the web images are labeled by users according to their prior knowledge, so that inaccurately annotated tags and irregularly mislabeled patterns are very common in such images (see Figure 4.1). Directly using a dataset with a high proportion of noisy labels can even degrade the performance of finely-trained CNN models [71, 134, 135]. Thus, it is highly desired to design a network that is able to mitigate the impact of noisy labels.

FIGURE 4.1: Examples of noisy labels. The image annotations that are manually labeled by amateurs or automatically generated by a machine are not reliable. Worse, the noisy labels and the number of mislabeled images are not specified.

The main challenge of noisy data learning is that we do not know which image(s) is incorrectly labeled, neither the total number of such images. There are many methods studying the problem. These methods can be roughly grouped into two kinds, either to improve the quality of training data, i.e., removing or relabeling the incorrect labels, or to design a noisy-robust models. We consider to combine both methods and propose to purify the noisy data. Then we use the resulting dataset to train our proposed noisy-robust CNN model.

4.2 Noisy Labels Purifying

Although the given labels may be incorrect, the contents of images hold the truth. The feature similarity of two images in the same class is still strong even though one of them is wrongly labeled; therefore it is possible to correct the incorrect labels of some images by evaluating their similarities to all classes. Since the given labels may be incorrect, in this section, we propose to use unsupervised common pattern discovery to mine the representative images of each class, and the similarities of other images to them are used as a criterion for relabeling.

4.2.1 Common Pattern Mining from Noisy Images

Assume that in each class the correctly labeled images make up the majority, and the other images with the same given labels could contain objects of any other classes in the given dataset. Let $\{(x_n, \tilde{y}_n)\}_{n=1}^N \subset \tilde{D}$ denote the noisy dataset, where x_n is the n -th training sample and $\tilde{y}_n \in \{1, 2, \dots, C\}$ is the given label (may be incorrect). The first step is to find these representative images of each class. To this end, we apply the *S-UOL* model proposed in Chapter 3 to the images of the same given class so as to identify the images containing the dominant patterns. Following Section 3.2.4, we employ [6] to extract image features, resulting a 4096-dimensional feature vector for each image. For images with the same given labels, we calculate their cosine similarities to build the similarity matrix. Different from the typical *S-UOL* model where both similarities and node confidence are used, we only consider the similarities among images in this section; the node confidences are all set to be 1. The results are a subset of images that appear most frequently in corresponding classes. We conduct the process respectively for each of the C classes to find out the most common patterns, denoted as $\{S_i\}_{i=j}^C$, S_j is common image set for class j , containing N_j images.

4.2.2 Noisy Image Relabeling and Purifying

After obtaining the common/representative images of each class, we assign a label to each image as its nearest class. In particular, the similarity of an image x_n to a class is evaluated by its average similarity to all representative images:

$$\langle x_n, S_j \rangle = \frac{1}{N_j} \sum_{i=1}^{N_j} \langle x_n, S_{i,j} \rangle, \quad (4.1)$$

where $S_{i,j}$ denotes the i -th representative image of j -th class. The label assigned as:

$$y'_n = \arg \max_j \langle x_n, S_j \rangle \quad (4.2)$$

We term y'_n as “content label” since it is predicted based on content similarities. If the content label of an image is consistent with the given label, we consider the label to be reliable. Note that, this does not mean the image is correctly labeled. On the other hand, if the content label is different from the given label, we relabel the image based on its content only if its similarity to content class overwhelms the similarities to other classes. That is, for an image x_n , $y' \neq \tilde{y}$, if $\frac{\langle x_n, S_j \rangle}{\sum_{j=1}^N \langle x_n, S_j \rangle} > st$, we then replace the given label with the content label. We term the similarity threshold st used to determine whether to relabel or not as relabeling threshold. We fix it as 0.8 in our experiments, and ablation studies on various values also presented in Section 4.4.

This purifying process does not guarantee the images with wrong labels can be all corrected. However, in the experiments we show that it could improve the number of images correctly labeled, which definitely contributes to the final performance of the designed learning models from these noisy data.

4.3 Attention-Aware Noise Learning

In this section, we present an attention-aware noise learning framework (A^2NL) which can be employed by normal networks to handle classification tasks in the presence of noisy labels. Specifically, we firstly introduce a Noise-Attention model with multiple noise-specific units to learn the noisy distributions of the given image set. The attention network is trained with the proposed recursive learning strategy. That is, the attention network (except the first one) will be iteratively trained under the training supervisions updating dynamically with its two components: a) the given training labels; and b) the outputs from the previous round. Figure 4.2 shows the recursive learning process of the attention network. Compared with directly combining multiple networks, the proposed recursive learning strategy could

FIGURE 4.2: An overview of the proposed attention-aware noisy learning approach. Firstly, a Noise-Attention model is introduced, which consists of multiple noise-specific units (NS unit) with each paying attention to a specific noisy distribution. The attention network is trained using the proposed recursive learning strategy. At each recursive round, the training supervisions combined from the given labels and the outputs from the previous round will guide the network learning in the current round.

integrate the learned knowledge of well-trained networks, while without consuming much more computing resources.

4.3.1 Noise-Attention Model

A typical classification network is usually followed by a softmax layer in the end, the outputs of which indicate the class confidences predicted by the network. For an input sample $\{x_n, y_n\}$, the output of softmax layer is usually written as: $\mathbf{p} = \{p(y = j|x, \theta)\}_{j=1}^C$, where C is the number of classes, and θ represents the CNN parameters. The learning objective is to minimize the negative log likelihood over N samples:

$$L(\theta) = -\frac{1}{N} \sum_{n=1}^N \log p(y = y_n | x_n, \theta) \quad (4.3)$$

Different from traditional object recognition tasks, we consider a more challenging situation in which the image sets may contain noisy labels. We still use $\{(x_n, \tilde{y}_n)\}_{n=1}^N \subset \tilde{D}$ to denote the noisy dataset, but some of the labels $\tilde{y}_n \in \{1, 2, \dots, C\}$ are revised based on the purifying strategy proposed in Section 4.2. The true label is denoted by $y_n^* \in \{1, 2, \dots, C\}$, which is treated as a latent variable. Then the conditional distribution of the given label on the true label of a sample is

FIGURE 4.3: Illustration the definition of the noise-specific unit.

described by $Q = \{q_{ji}\}_{i,j=1}^C$, $q_{ji} := p(\tilde{y} = j | y^* = i)$, where Q is a $C \times C$ matrix with the sum of each column equal to 1. Different from [70, 72], which describe the noise in the whole dataset with only one noisy distribution, the proposed Noise-Attention model contains multiple noisy units with each paying attention to a specific noisy distribution, parameterized by matrix Q_m :

$$\begin{cases} Q_m = I & \text{if } m = 1 \\ Q_m \neq I & \text{if } m > 1 \end{cases} \quad (4.4)$$

where I is an identity matrix, Q_m is the noise distribution matrix in the m -th unit, $m = \{1, 2, \dots, M\}$, M is the total number of noise-specific units. As the two dog samples shown in Figure 4.3, the furry one is easy to be labeled as a cat and the big one tends to be mislabeled as a horse, so that multiple label noise distributions are necessary for catering different prototypes. Here, $Q_1 = I$ is fixed to represent the noisy distribution of clean samples, while other noise-specific units will update during training to learn the noisy distributions.

In order to apply the proposed Noise-Attention model into CNNs, we model the label noise by a constrained linear layer, with each weight matrix of a linear layer representing a noisy distribution Q_m . The noisy units contained in the Noise-Attention model are implemented by parallel linear layers, fully-connected layer in CNNs. Figure 4.2 highlights the proposed Noise-Attention model. For an input sample $\{x_n, \tilde{y}_n\}$, it will first pass through a typical CNN, which we call base model,

and then the proposed Noise-Attention model sequentially. The intermediate outputs from base CNN will pass through all the M noisy units in the model and thus generate M outputs: $\{\mathbf{p}_m\}_{m=1}^M$, $\mathbf{p}_m \in \mathbb{R}^{C \times 1}$. Instead of using all noisy units in the NA model, we only use the one that maximizes the prediction confidence, i.e., $Q_{max} = \arg \max_Q p(y = \tilde{y}|x_n, \theta, Q_m), m = 1, \dots, M$. Therefore, the confidence of an input x_n being predicted as class j by the attention network is given by:

$$p(y = j|x_n, \theta, Q_{max}) \quad (4.5)$$

Then the negative log likelihood loss for the attention network can be written as:

$$L(\theta, Q) = -\frac{1}{N} \sum_{n=1}^N \log p(y = j|x_n, \theta, Q_{max}) \quad (4.6)$$

We employ stochastic gradient descent (SGD) with back-propagation technique to optimize the parameters in the attention network. Since the Noise-Attention model consists of linear layers (implemented by fully-connected layers without bias term), the gradients can pass through them to the base network. Before learning the Noise-Attention model, we first train the base network with noisy data to its optimal so that the noisy information could be well absorbed. Then the noise-specific units parameterized by $\{Q_m\}_{m=1}^M$ are added gradually to the base model, with each being initialized as identity (matrix), $Q_m = I$. A new noise-specific unit will be involved and updated with larger decay once the performance improvement is limited with the current unit. By this training strategy, the noisy information absorbed by the base network in the first dozens of epochs is expected to gradually distill to the Noise-Attention model. In addition, the noise-specific units are forced to learn diverse noisy distributions with sequentially adding strategy and different learning decay values, which are experimentally verified in Section 4.4.

The goal of the proposed method does not end with predicting the noisy labels $\tilde{y}$ or learning the noisy distributions, instead it is to predict the true labels given the image sets with noisy labels. Recall that the true label y^* is regarded as a latent

variable, its relation with the noisy labels and noisy distributions can be expressed as:

$$\begin{aligned}
& p(\tilde{y} = j | x_n, \theta, Q_{max}) \\
&= \sum_{i=1}^C p(\tilde{y} = j | y^* = i) p(y^* = i | x_n, \theta) \\
&= \sum_{i=1}^C q_{ji} p(y^* = i | x_n, \theta)
\end{aligned} \tag{4.7}$$

where $p(\tilde{y} = j | x, \theta, Q_{max})$ and $Q_{max} = \{q_{ji}\}_{i,j=1}^C$ represent the predictions of the attention network and the noisy distribution. Obviously, $p(y^* = i | x_n, \theta)$ is the prediction of base network. Therefore, if the given training labels are correctly predicted (i.e., $p(\tilde{y} = \tilde{y}_n | x_n, \theta, Q_{max}) = 1$) and the noisy distributions are exactly learned (i.e., $q_{\tilde{y}_n, y_n^*} = 1$, $q_{j \neq \tilde{y}_n, i \neq y_n^*} = 0$), then theoretically the base network is able to predict the true labels accurately [70]. Although the Noise-Attention model will introduce more parameters in training phase, it is similar to traditional classification network in inference with the outputs of the base network being used to evaluate.

4.3.2 Recursive Learning

In this section, we put forward a CNN training strategy, termed recursive learning, which could take advantage of the predictions from CNNs to further boost the learning ability of our attention network. It is inspired by [77, 78, 136] that the predicted class confidences of a well-trained classifier usually contain rich information. The predicted confidence can not only indicate the object class of the input image, but also reflect the relations among the classes. For example, if a cloth sample is predicted as an orange one with the confidence of 80%, a brown one with 15%, and 5% for other classes. The biggest figure (80%) advocates that the input image contains a cloth in orange, and other figures suggest that the orange is highly possible to be mixed up with the brown.

The key idea of our proposed method is to fine-tune the attention network by leveraging the high-level knowledge from a well-trained network, so that it can make predictions similar to the ensemble of the two. In addition, the proposed recursive learning is in an iterative manner, with shared network architectures and dynamically updating training supervisions in each round, making the final network a composite model of all previous networks. Specifically, the training supervisions in current round consist of two parts, 1) the fixed given training labels, and 2) the outputs of the attention network from the previous round. The network is initialized with parameters from the previous round, and is able to bootstrap itself with the updated training supervisions. The process continues until the improvement over the previous round is negligible, then the model in the last round will be used for inference. Figure 4.2 outlines our proposed recursive learning procedure. Compared with combining the models in all rounds for inference, our recursive learning strategy is smaller in size to store and faster in runtime to evaluate.

Sample-Wise Summation. Inspired by [78], the network trained by the combined training supervisions can achieve a performance similar to an ensemble model combining two networks. We thus integrate the high-level knowledge in a well-trained attention network with the given labels as the new training supervisions. The combination will be applied to guide the attention network training in the current round. For a given sample $\{x_n, \tilde{y}_n\}$, $p(y = j|x_n, \theta^{t-1}, Q^{t-1})$ represents the confidence of being classified as class j by the network from the $(t - 1)$ -th round. Let $\mathbf{s}_n = \{s_{n,j}\}_{j=1}^C$ denote the combined training supervisions of sample x_n , then the weighted combination is formulated as:

$$s_{n,j}^t = \alpha^t \cdot \mathbf{1}_{n,j} + p(y = j|x_n, \theta^{t-1}, Q_{max}^{t-1}) \quad (4.8)$$

where $\mathbf{1}_n$ is a binary vector with only one non-zero element indicating the given label of the input x_n , and the subscript j denoting its j -th element. Thus, the

above formulation can be rewritten as:

$$\begin{aligned} s_{n,\tilde{y}_n}^t &= \alpha^t + p(y = \tilde{y}_n | x_n, \theta^{t-1}, Q_{max}^{t-1}) \\ s_{n,j \neq \tilde{y}_n}^t &= p(y = j | x_n, \theta^{t-1}, Q_{max}^{t-1}) \end{aligned} \quad (4.9)$$

$\mathbf{s}_n \in \mathbb{R}^{C \times 1}$ is then normalized with its elements summing to 1. Also, only the noisy unit that maximizes the confidence of correctly classifying, Q_{max} , is used. The coefficient α is manually fixed as 0.8^t . It is decreasing with more rounds as the network is expected to be more reliable. Compared with directly combining two networks, such a combination could also integrate the knowledge in pretrained network while without introducing much more parameters.

At the first round (denoted by A^2NL in Figure 4.2), the attention network is trained with given labels as in traditional classification task, and all parameters of the network are trained from scratch. From the second round, the networks are initialized by parameters learned in the previous round, and the training supervisions are composed of the given labels and the outputs of previously trained network. Figure 4.2 illustrates the recursive learning process. The network at the first round is trained with objective function presented in Equation (4.6). From the second round, the objective function is to minimize the negative log likelihood with the updated training supervisions:

$$L(\theta, Q) = -\frac{1}{N} \sum_{n=1}^N \sum_{j=1}^C s_{n,j}^t \log p(y = j | x_n, \theta^t, Q_{max}^t) \quad (4.10)$$

Although the proposed recursive learning method consumes more computing resources in the training phase, during inference the complexity does not increase as only the base network at the last round is used. Since in each recursive round the predictions from the previous round are integrated as supervisions, the learning ability can be distilled gradually, which yields the last network that makes predictions similar to the ensemble of models in all rounds [77]. Thus, compared with existing works [63, 71], which require two independent networks in both training and inference, our proposed method is more efficient and effective during inference.

FIGURE 4.4: Cloth samples with multiple attributes. The first row shows samples from training set (red ones are mislabeled), and second row from testing set.

4.3.3 Application to multi-attribute classification

The proposed Noise-Attention model which is implemented as a linear layer can be flexibly plugged into any standard network architecture, and the proposed recursive learning strategy can be adopted for general network training. To show this, we extend our proposed approach to multi-attribute learning, i.e., cloth attributes classification, which is more challenging than a single attribute classification task, especially in the noisy scenario. Figure 4.4 shows some examples from the cloth dataset [137], there are five attributes for each sample (Clothes Category, Color, Collar Shape, Sleeve Length, and Button). As some attributes, such as Color and Sleeve Length, are easy to be recognized in most cases and others, such as the type of Button, can be quite difficult as the noisy levels of the five attributes may vary a lot. Instead of training a whole CNN model for each attribute as in [138], we append multiple Noise-Attention models for various attributes on top of the shared base network. Let $\{x_n, \mathbf{y}_n\}$ be a sample with K attributes, i.e., $\mathbf{y}_n \in \mathbb{R}^{K \times 1}$, and the k -th attribute has C_k classes, i.e., $\mathbf{y}_{n,k} \in \{1, 2, \dots, C_k\}$. Then for each attribute we employ M noise-specific units parameterized by $Q \in \mathbb{R}^{C_k \times C_k}$. The recursive learning procedure is adopted similarly to that of the single attribute classification, in which the training supervisions are updated attribute-wisely. The network cost function is calculated independently for each attribute, for example,

the negative log-likelihood loss of the c -th attribute in the first round/stage is:

$$L_k(\theta, Q^k) = -\frac{1}{N} \sum_{n=1}^N \log p(y = \tilde{\mathbf{y}}_{n,k} | x_n, \theta, Q_{max}^k) \quad (4.11)$$

The cost function for the network with combined training supervisions is similar to the one defined in single attribute task. The only difference is that for the multi-attribute task, there are K classifiers taking as input the shared CNN features, each of which is followed by a Noise-Attention model. The back-propagation follows the same pipeline with the single attribute task.

4.4 Experiments

To evaluate the proposed purify process, Noise-Attention model and the recursive learning strategy, we conduct experiments on two popular datasets: CIFAR-10 [139] and SVHN [140] for single attribute classification and a real-world clothes dataset [137] for multiple attributes classification. Since the first two datasets are noise free, we randomly select and flip the labels of some training samples. Let ρ denote the initial noisy level of the training set, i.e., the percentage of mislabeled images in the training set. For the manually flipped datasets, we firstly purify the labels by our proposed pre-processing strategy, the noisy levels before and after relabeling are listed in Table 4.1. For the real-world dataset, since the images are collected from websites with user-supplied tags, it naturally contains a certain level of mislabeled samples. However, we do not know the true noisy levels, so they are not explicitly listed.

Baselines: Several recently proposed noisy label learning methods are chosen as the baselines:

1. Bottom-up [70]: A CNN model contains a single noisy layer on the top to learn noisy distribution;

2. Joint CNNs [71]: A combined model consists of two CNNs;
3. Dropout-Reg [141]: A regularization scheme is proposed and applied to the output of CNN to prevent it from learning the noisy labels directly;
4. SEC-CNN [142]: A confidence policy is introduced to correct the wrong label by switching between the label of the sample and the max-activated output neuron of the CNN;
5. Iterative [63]: A Siamese network is designed to enforce clean labels and noisy labels to be dissimilar, which is learned in an iterative fashion;
6. GeneralizedCCE [143]: A set of noise-robust loss functions which can be applied to any CNN architectures and algorithms.
7. ResistanceNN [144]: Inspired by k-NN classifier, Drory *et al.* demonstrate that CNNs also predict the label of a sample based on its neighbors in the training set, which is applied to the problem of noise label classification.
8. BundleNet [145]: A sample-bundle module is designed to utilize sample correlations by constructing bundles of samples class-by-class. The bundle module could be applied to any CNN architectures by replacing the traditional input layer with it.
9. Distillation [146]: A unified distillation framework is designed to reduce the influence of label noise by “side” information, including a small clean dataset and label relations in the knowledge graph.
10. Base model: the vanilla CNN.

The proposed attention network and the recursive learning strategy are implemented on Caffe [147]. We resort to Caffe’s CIFAR-full model as the base network for the first two datasets, and DARN [137] for real-world cloth dataset. To save space, we abbreviate the network containing Noise-Attention model as “NAtt.”, the attention network trained using recursive learning as “NAtt. + Rec.”, and the attention network trained using traditional training process as “NAtt. + Iter.”.

4.4.1 Manually Flipped Noise

Dataset Details: The CIFAR-10 dataset is introduced in 2009 for representation learning from tiny images. It consists of 60k 32×32 colour images in 10 classes, with 6k images per class. There are 50k training images and 10k test images. The 10 classes are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. The dataset has been widely used in the past 10 years for image classification and object detection. The Street View House Numbers (SVHN) Dataset is introduced in 2011, which contains house numbers from Google Street View images. It consists of 600k 32×32 digit images in 10 classes, one for each digit. Following the common practice, we use about 100k images for training and 26k for testing. Since the two datasets are correctly labeled by experts, we randomly select a certain proportion of samples and flip them into the other categories to evaluate the proposed method under random noise. The noisy level, i.e., the proportion of mislabeled images, ranges from 0.1 to 0.7.

Relabeling by Various Thresholds. We conduct experiments using different values of similarity threshold for relabeling. The noisy levels before and after relabeling are shown in Table 4.1. Theoretically, too large relabeling threshold limits the number of samples corrected, while too small relabeling threshold could introduce more noise if the original noisy level is large. It can be seen from Table 4.1 that more noises are introduced to training samples in some cases when the relabeling threshold is too small. We fix the relabeling threshold as 0.8 as it performs averagely well on both datasets.

Influence of Number of Noise-Specific Units. The number of noise-specific units contained in Noise-Attention model is adjustable according to the dataset and noisy level. In this part, we will illustrate the influences of different numbers of noisy-specific units. As stated in Section 4.3.1, a noise-specific unit will be added to the NA model once the training loss converges. Before that, the base model of CIFAR-10 is trained for 30 epochs, and that of SVHN for 20 epochs so that the noisy information is fully learned by the base networks. Then we train the networks

TABLE 4.1: Noisy levels (%) before and after relabelling.

(a) CIFAR-10							
relabelling thresh.	Initial Noisy Level						
	10(%)	20(%)	30(%)	40(%)	50(%)		
0.9	6.9	14.2	25.2	34.3	47.7		
0.8	6.4	13.1	24.3	34.7	46.0		
0.7	6.7	13.0	28.7	35.9	47.4		
0.6	7.4	15.8	29.0	37.3	49.3		
0.5	9.2	17.4	31.2	39.6	52.1		

(b) SVHN							
relabelling thresh.	Initial Noisy Level						
	10(%)	20(%)	30(%)	40(%)	50(%)	60(%)	70(%)
0.9	3.5	12.2	21.7	33.5	44.8	51.3	66.2
0.8	3.2	11.4	20.8	31.6	43.7	52.1	65.7
0.7	2.8	13.1	22.9	32.1	45.1	54.5	67.8
0.6	3.4	15.8	24.3	35.7	48.1	61.3	69.1
0.5	4.7	17.6	26.8	41.1	50.2	63.4	72.5

while adding noise-specific units to the NA model one by one until the performance improvement is limited.

Figure 4.5 shows the test errors of the proposed attention network with different numbers of noisy unit $\{Q_m\}_{m=1}^M$. Recall that in our definition Q_1 is fixed as I , so the attention network reduces to the base model if it contains only one noise-specific unit. From the two figures, one can see that adding more noise-specific units has slight improvement in performance when noisy level is low, such as $\rho = 0.1$ on CIFAR-10 and $\rho = 0.1, 0.2$ on SVHN. This is reasonable since CNN models intuitively have strong learning ability to handle a small portion of noisy labels. However, the noisy tolerance capacity of the base model decreases sharply with the increasing number of noisy samples. Although our proposed noise-attention network also degrades with high-level noise, the superiority over the base model is remarkable, e.g., test errors reducing 6% on CIFAR-10 at $\rho = 0.4$. In general, the more noise-specific units are added, the better the classification results will be. However, the improvement slows down. Hence, we use $M = 5$ noise-specific units for CIFAR-10 and $M = 4$ for SVHN in the following experiments to evaluate the recursive learning strategy.

FIGURE 4.5: Test errors (%) on CIFAR-10 and SVHN in terms of number of Noise-Specific Units in Noise-Attention model.

Influence of Number of Recursive Rounds. Although the attention network has achieved satisfactory performance, it can be further boosted by the recursive learning strategy. The outputs of the well-trained attention network with multiple noise-specific units will be combined with the given labels as the new training supervisions for recursive learning process. The number of noise-specific units is fixed as $M = 5$ for CIFAR-10 and $M = 4$ for SVHN. The coefficient α for the given labels in t -th round is set as $\alpha = 0.8^t$. Instead of adjusting and learning Noise-Attention model as in the first stage, we focus on the whole network learning in the recursive learning stage.

Figure 4.6 shows the test errors changing with the number of recursive rounds. The results of the network in the first stage are denoted by $t = 0$. The test errors decrease dramatically after one round, and the trend slows down with more rounds. For CIFAR-10, the proposed recursive learning strategy has an average improvement of about 2% after training four rounds. For SVHN, after four rounds, the model trained at 50% noisy level even performs better than the model with only 10% mislabeled samples present in the training set.

Visualization of Noise-Specific Units. We visualize the learned noise-specific units in Figure 4.7 for CIFAR-10 at $\rho = 0.3$ and SVHN at $\rho = 0.5$. Figures in the first column represent Q_1 which is fixed as an identity matrix. The other columns show the learned noise-specific units, with the diagonals indicating the

FIGURE 4.6: Test errors (%) on CIFAR-10 and SVHN in terms of number of recursive rounds.

(a) CIFAR-10 at $\rho = 0.3$.

(b) SVHN at $\rho = 0.5$.

FIGURE 4.7: Visualization of noise-specific units. The first column represents $Q_1 = I$, the rest ones show the learned noise-specific units.

correctly labeled ratios. Due to the gradually learning strategy, the learned noisy distributions are diverse to each other so that different noisy information in training samples can be better modeled.

Comparison with State-Of-The-Art. To fairly evaluate our proposed approach, we compare the results with related literature in Table 4.2. It can be seen that, the purifying strategy works well at lower noisy levels for both datasets. The reason is simple – with heavier noisy levels, the representative images are more difficult to select, resulting in unreliable class similarities. The final performance listed on the table is achieved with $M = 5$ noisy units for CIFAR-10 dataset and

TABLE 4.2: Test errors in the presence of noisy labels.

(a) CIFAR-10.							
Initial Noisy Level	10%	20%	30%	40%	50%		
Relabeling	6.4%	13.1%	24.3%	34.7%	46.0%		
Base model	20.27	23.49	26.53	30.09	30.92		
Bottom-up [70]	19.74	20.88	22.83	25.2	26.4		
Joint CNNs [71]	19.6	20.5	22.8	25.0	26.47		
Dropout-Reg [141]	-	-	25.4	-	31.28		
SEC-CNN [142]	21.99	26.75	30.09	38.35	44.72		
Iterative [63]	-	18.64	-	21.85	-		
GeneralizedCCE [143]	18.39	19.12	19.89	21.88	-		
ResistanceNN [144]	18.7	19.2	20.0	26.5	56.0		
BundleNet [145]	11.78	-	19.66	-	43.26		
Distillation [146]	15.24	17.60	19.51	21.89	24.16		
NAtt.	18.97	19.92	21.05	23.7	25.5		
NAtt. + Iter.	18.27	19.95	21.1	23.28	25.1		
NAtt. + Rec.	16.95	17.97	19.13	22.02	23.83		

(b) SVHN.							
Initial Noisy Level	10%	20%	30%	40%	50%	60%	70%
Relabeling	3.2%	11.4%	20.8%	31.6%	43.7%	52.1%	65.7%
Base model	6.8	7.35	7.85	8.8	10.4	16.0	22.0
Bottom-up [70]	6.77	7.2	7.6	8.0	8.7	9.2	10.0
Joint CNNs [71]	6.75	7.08	7.41	7.67	8.7	9.21	9.79
Dropout-Reg [141]	7.04	7.28	7.91	8.73	10.3	15.27	18.6
Distillation [146]	5.54	6.21	6.27	6.53	6.71	7.5	8.01
NAtt.	6.7	6.98	7.19	7.43	7.92	8.28	8.8
NAtt. + Iter.	6.83	7.1	7.21	7.37	7.76	8.14	8.35
NAtt. + Rec.	5.44	6.2	6.31	6.4	6.8	7.42	7.96

$M = 4$ for SVHN dataset, $t = 4$ recursive rounds for both datasets. To validate the effectiveness of the proposed recursive learning method, we list the results achieved by an attention network learned using traditional training process but with the same epochs as in “NAtt. + Rec.,” which we denote as “NAtt. + Iter.”.

Table 4.2(a) lists the test errors on CIFAR-10 with the considered methods at different label noisy levels. All results have been reported in the corresponding publications, except Distillation [146] which is implemented based on the released code. Our proposed noise-attention network (“NAtt”) outperforms the majority

of baselines except the Iterative [63]. After four-rounds of training using recursive learning strategy (“NAtt+Rec.”), our proposed method beats Iterative [63] by about 1.4% at $\rho = 20\%$ and about 0.8% at $\rho = 40\%$. While the performance of GeneralizedCCE [143], ResistanceNN [144], and Distillation [146] is on par with our method at the low level of label noise. These methods are inferior to ours under the high proportion of label noise. Note that the Distillation [146] even uses extra clean samples during training, and the size of the clean dataset is one-quarter of the training set on both CIFAR-10 and SVHN. The results achieved by “NAtt. + Iter.” are similar to these of “NAtt.” as the network in “NAtt. + Iter.” is exactly the one in “NAtt.” except being trained with more epochs; however more training epochs do not always lead to better performance. Although sharing the same architecture and running with equal epochs, the attention model trained by “NAtt+Rec.” performs favorably against that by “NAtt. + Iter.” with averagely 2% improvement, which demonstrates the effectiveness of our proposed recursive leaning method.

Table 4.2(b) lists the results on SVHN dataset, among which the Joint CNNs [71], Dropout-Reg [141] and Distillation [146] are implemented by ourselves following their released code. With extra clean data, Distillation [146] performs similarly to our method. The performance obtained by Joint CNNs and Bottom-up is similar on this dataset, while it is superior over Dropout-Reg [141] and is inferior to the proposed attention network (“NAtt.”), especially at high noisy levels. More training epochs on attention network (“NAtt. + Iter.”) even degrade the model at low-level noise due to network overfitting. For both datasets, the base model degrades severely with the increasing number of noisy samples, while our noise-attention network is more stable and can be further improved with the recursive learning method.

TABLE 4.3: The statistics of the real-world clothing dataset used in our experiments.

Attribute categories	Examples (total number)
Clothes Category	T-shirt, Skirt, Leather Coat ... (20)
Clothes Color	Black, White, Red, Blue ... (9)
Collar Shape	Round, Lapel, V-Neck ... (5)
Sleeve Shape	Puff, Raglan, Petal, Pile ... (10)
Clothes Button	Double Breasted, Pullover, ... (12)

4.4.2 Real-world Clothing Noise

We further evaluate the considered methods on a real-world cloth dataset, which is an open dataset introduced in [137], whose images are crawled from several popular online shopping websites. The dataset contains not only images taken and labeled by professionals, but also user-supplied images that are captured and labeled under uncontrolled conditions. Specifically, each image has 9 semantic attribute categories, and there are more than a hundred attribute labels in total. We select 5 attribute categories in our experiment. The total number of attribute labels is 56, and the statistics are listed in Table 4.3. Since the labels are user-supplied, it is highly possible that the noisy distributions of different attribute categories will vary a lot. We select 118,057 images from the dataset for training, and another 2000 images which are manually purified by us for testing. The network from [137] is chosen as the base model, which is trained for 30 epochs till convergence. Then the noise-specific units is gradually added to the Noise-Attention model one by one until the performance improvement is limited. There are 6 noise-specific units contained in each NA model before being fine-tuned using recursive learning method, which lasts for another 5 recursive rounds before stopping training.

Figure 4.8 shows the test errors (%) changing with the different numbers of noise-specific unit $\{Q_m\}_{m=1}^M$ and at different rounds of the recursive learning strategy. The “.ALL” in title means that the samples are true positive only if the 5 attribute

FIGURE 4.8: Test errors (%) on real-world cloth dataset in terms of number of noise-specific units in Noise-Attention model and number of recursive rounds.

categories are all correctly predicted. As the noisy distributions of real-world images are rather complicated, the proposed attention network enjoys significant improvement over the base model by about 4% with $M = 6$. The performance can be further enhanced by our recursive learning strategy. Figure 4.8(b) depicts the decreasing tendency of the test errors with the increasing number of recursive rounds. After only one round, the performance is improved by more than 1%. We stopped training at $t = 5$ for the improvement over the previous round is negligible. Compared with the Bottom-up model [70], which only sets a single noisy layer, the proposed noise attention network achieves 2% improvement. After being fine-tuned using recursive learning strategy, the improvement rises to 4%.

Table 4.4 lists the text errors (%) achieved by considered methods on the individual attribute category, and “ALL” means that the five attribute categories of a sample are all predicted correctly. The results of Bottom-up [70], Joint CNNs [71], Dropout-Reg [141] and Distillation [146] are obtained using their released codes. From the table, we can see the performance achieved on different attribute categories varies a lot even using the same method. The possible reason is that some attribute categories are easy to be labeled, and some could be sensitive to the environment. Among the considered methods, only the Joint CNNs [71] performs slightly better than our attention network, but its performance is far behind the

TABLE 4.4: Test errors (%) of the considered methods on clothes images.

SVHN	Clothes Category	Clothes Color	Collar Shape	Sleeve Length	Clothes Button	ALL.
Base model	9.27	30.19	33.02	17.56	31.14	40.39
Bottom-up [70]	8.5	26.78	32.2	15.1	30.73	39.17
Joint CNNs [71]	6.93	23.88	29.7	12.04	27.7	36.46
Dropout-Reg [141]	8.98	27.26	33.4	15.81	30.2	38.47
Distillation [146]	8.31	23.74	28.57	13.35	26.88	35.19
NAtt.	7.1	24.8	30.38	12.85	28.6	37.29
NAtt. + Iter.	7.13	24.16	30.44	12.7	27.96	36.14
NAtt. + Rec.	6.4	22.0	29.07	11.0	26.8	34.8

network trained using our recursive learning strategy. Note that [71] is trained using an extra number of clean data, but [70], [141] and our attention network are all trained using noisy data. Although the Distillation [146] performs on par with our “NAtt. + Rec.”, it requires extra clean data for training, an extra set of 2000 images with correct labels is used for pretraining the Distillation model. The proposed method and Joint CNNs outperform Bottom-up by a large margin as it favors uniformly distributed noise. Similar to the results in manually flipped label noise scenario, the model trained in “NAtt. + Iter.” performs similarly to “NAtt.”, while is significantly inferior to “NAtt. + Rec.”. Although the three baselines and the base model are more efficient during training, the testing phase of the proposed attention network trained with recursive learning strategy is similar as traditional classification network with only the base network being used to evaluate.

4.5 Conclusion

In this chapter, we address the noisy learning problem from two aspects, to improve the quality of training set and to build a robust learning model. On the one hand, based on the similarities of an images to the common patterns of a class, we design criterion to correct the unreliable labels. On the other hand, we propose an attention-aware noisy label learning approach to address the problem of image classification in the presence of label noise. Firstly, a Noise-Attention model is designed to learn the diverse noisy label information in the training set. It consists

of multiple noise-specific units, each of which focuses on a specific noisy label distribution. To strengthen the learning ability of the attention network, we further introduce a recursive learning strategy which could integrate the high-level knowledge from another network without introducing more parameters. In the learning process, a well-trained attention network is able to bootstrap itself by the combined training supervisions of given labels and outputs from the previous round. To validate the proposed methods, we conduct experiments on both synthesized label noise datasets and real-world cloth dataset with multiple attributes. The ablation studies on various numbers of noise-specific units and numbers of recursive rounds show the effectiveness of the Noise-Attention model and recursive learning strategy. The extensive experiments on images with deliberately flipped label noise and real-world label noise show the generalization of the proposed approach.

Chapter 5

Pruning 3D Filters for Accelerating 3D ConvNets

5.1 Introduction

Convolutional neural networks (CNNs) have evolved to become the state-of-the-art techniques for many computer vision and multimedia tasks. The performance is generally enhanced with deeper and wider network architecture. However, as networks expand, their memory and computational cost also increase rapidly, making them difficult to be deployed for applications with computing or power constraint, such as mobile devices. Unfortunately, works on accelerating ConvNets by reducing FLOPs are limited and they mainly focus on 2D ConvNets [8, 148, 149].

In this chapter, we tackle the problem of accelerating 3D ConvNets from the view of common pattern discovery. 3D ConvNets are widely applied to action recognition [150, 151] and video understanding [152–154]. Compared with 2D ConvNets, accelerating 3D ConvNets is more demanding for 2 reasons: 1) video data are inherently more redundant than individual images, thus it provides great potential to select effective filters from a large group of candidates; 2) reducing video data processing time is even more critical due to the huge video data size. The

FIGURE 5.1: Filter pruning in 2D ConvNets has shown excellent performance due to the redundant parameters in ConvNets, even though the inputs for 2D ConvNets are unrelated images. Will 3D ConvNets, whose inputs are sequential frames, be even more redundant?

main difference from 2D ConvNets is that convolution filters of 3D ConvNets are 3-dimensional, i.e., (k, k, d) with the third dimension d denoting the filter temporal depth, so that the temporal information could be captured. Figure 5.1 shows the difference between 2D ConvNets and 3D ConvNets.

5.2 Our Method

Notations. Consider a well-trained 3D ConvNets with N layers whose weights of ℓ -th layer are parameterized by a 5-dimensional tensor $W^\ell \in \mathbb{R}^{n_{\ell+1} \times n_\ell \times k \times k \times d}$, where $n_{\ell+1}$ is the number of output channels, n_ℓ represents the number of input channels, and $k \times k \times d$ is the size of a 3-dimensional filter cube. For better illustration, we arrange the 5-dimensional tensor as a matrix, denoted as $\mathcal{F}_\ell$, with each element corresponding to a filter cube and each column corresponding to an entire filter, $\mathcal{F}_{\ell,i}$ (see Figure 5.2). Our objective is to delete the redundant filters in ℓ -th convolutional layers (gray column in Figure 5.2) and the corresponding feature maps (in gray) in $(\ell + 1)$ -th layer. The resulting pruned network is expected to be computationally less expensive with minimum accuracy drop.

FIGURE 5.2: Illustration of our proposed 3D filter pruning. Given a well-trained 3D ConvNets, the duplicate and less informative filters are pruned (highlighted by gray in $\mathcal{F}_\ell$). Consequently, the following feature maps can also be removed, so can the filter cubes corresponding to the removed feature maps.

5.2.1 Filter Pruning via Determinantal Point Processes

Our goal is to select a subset of filters that are informative yet discriminative to each other, which is consistent with the goal of Determinantal Point Processes (DPPs) sampling where items corresponding to large magnitudes are preserved and the ones presenting strong correlations are suppressed. To incorporate the properties of DPPs with our goal of selecting informative and non-duplicate filters, we regard each filter as an item and formulate the subset selection problem as DPPs sampling.

We start from single layer pruning, e.g., ℓ -th layer, and then generalize it to multiple layers across the given network. The index ℓ is sometimes omitted if it does not cause any confusion. Let $\mathbf{q}_i$ denote the informativeness factor of the i -th filter $\mathcal{F}_{\ell,i}$, and the marginal matrix $L \in \mathbb{R}^{N_{\ell+1} \times N_{\ell+1}}$ is calculated from $L_{ij} = \mathbf{q}_i S_{ij} \mathbf{q}_j$, where S_{ij} is the similarity between i -th and j -th filters. Therefore, the marginal matrix L contains all critical information needed to evaluate a subset, i.e., the individual filter importance and relationships among filters. Our goal is to find a subset Y containing filter index from a ground set $\mathcal{Y} = \{1, 2, \dots, N_{\ell+1}\}$ that maximizes the following objective function:

$$P(Y = \mathbf{y}) = \frac{\det(L_{\mathbf{y}})}{\det(L + I)}, \quad (5.1)$$

where I is the identity matrix, $L_{\mathbf{y}}$ is the principal minor (sub-matrix) with rows and columns selected according to the indices in $\mathbf{y}$.

The determinant function $\det(\cdot)$ brings the property of pairwise repulsion. To see that, we take a subset of two filters i and j as an example. We have:

$$P(Y = \{i, j\}) \propto \mathbf{q}_i^2 \mathbf{q}_j^2 (S_{ii}S_{jj} - S_{ij}^2). \quad (5.2)$$

If the filters i and j are the same, then $P(Y = \{i, j\}) = 0$ due to $S_{ij} = S_{ii} = S_{jj}$. Therefore, identical filters are less likely to be selected simultaneously. If only one filter is preferred, as $P(Y = \{i\}) \propto L_{i,i} = \mathbf{q}_i^2$, so that the filter with highest information are selected. The general version also holds: $P(Y = \mathbf{y}) \propto \left(\prod_{i \in \mathbf{y}} \mathbf{q}_i^2\right) \det(S_{\mathbf{y}})$. According to the first term $\prod_{i \in \mathbf{y}} \mathbf{q}_i^2$, the more informative filters are preferred, while similar filters are suppressed by the second term, i.e., the determinantal value calculated from the marginal matrix is proportional to the informativeness and is negatively correlated to the similarity. With the integrated formulation, by maximizing the determinantal objective function, the diverse yet informative filters are expected to be selected.

As a comparison, most existing works [83, 155] select convolutional filters using $\ell_{2,1}$ -norm. Specifically, each filter is concatenated as a vector, then the filters with high value of $\ell_{2,1}$ -norm are selected. Although this kind of method could identify filters with large nonzero values, the dense filters with small values are also likely to be selected. To avoid this situation, Li *et al.* [8] select filters with large ℓ_1 -norm so that the selected filters could be more sparse yet containing discriminative information. However, selecting filters only by this criterion may lead to duplicate filters. Our proposed method improves this by considering the relationships among filters together with the individual filter informativeness.

Optimization. As shown in many previous network compression studies [8, 82, 149], the accuracy drop is highly related with the compression ratio. High compression ratio generally leads to severe accuracy drop. Therefore, a controllable pruning ratio is necessary for adjusting to the requirements of different tasks. To this end, we resort to k -DPPs to optimize our objective function, where k is a manually fixed parameter denoting the size of the selected subset. Conditioning

on fixed sampling size k , the k -DPPs sampling can be formulated as:

$$P_k(Y = \mathbf{y}) = \frac{\det(L_{\mathbf{y}})}{\det(L + I)}, \quad \text{s.t.} \quad \|Y\|_0 = k. \quad (5.3)$$

The k -DPP sampling is able to balance the trade-off between network size and network performance by tuning the parameter k for intended subset size. A larger k typically leads to a lower compression rate with smaller performance drop, and vice versa. For a layer with n_ℓ filters, the total number of pruning masks is $\mathcal{O}(2^{n_\ell})$ and an exhaustive search is therefore infeasible even for a small size network. For example, the greedy search for a fixed ratio of pruning, say $\frac{k}{n_\ell}$, $k < n_\ell$, still requires $\mathcal{O}(\mathcal{C}_{n_\ell}^k)$. To address this problem, Anwar and Sung [156] propose to accelerate the sampling process by evaluating some random combinations. Such an ad-hoc strategy is highly dependent on these “random selected” combinations, and the best combination could be missed. In light of this, Li *et al.* [101] propose an efficient sampling optimization method specified for k -DPPs by constructing coresets for the ground set of items. By leveraging the fast k -DPPs, the complexity of our proposed sampling is reduce to $\mathcal{O}(n_\ell k^3)$. In our experiments, this sampling cost is negligible compared to the cost of training.

5.2.2 Constructing DPPs Marginal Matrix L

Recall that the marginal matrix L is constructed from the individual filter importance score $\mathbf{q}_i$ and the pairwise similarities S_{ij} . For the ℓ -th layer of a well-trained 3D ConvNets, the 5-dimensional tensor $W^\ell \in \mathbb{R}^{n_{\ell+1} \times n_\ell \times k \times k \times d}$ can be decomposed to $n_{\ell+1}$ filters $\{\mathcal{F}_{\ell,i} \in \mathbb{R}^{n_\ell \times k \times k \times d}\}_{i=1}^{n_{\ell+1}}$. Each filter is associated with a scalar $\mathbf{q}_i$ denoting the informativeness score and a vector Φ_i denoting the filter representation.

Informativeness Score $\mathbf{q}_i$. In the filter-level pruning, many existing works use $\ell_{2,1}$ -norm as a regularizer [82, 83, 155] and obtain promising results. Recent work [8] shows that there is no significant difference between the ℓ_2 -norm and the ℓ_1 -norm for filter selection since the important filters tends to have large values under

FIGURE 5.3: Visualization the distribution of importance score $\mathbf{q}_i$ of filters in all convolutional layers of C3D trained on Sports1M.

both measures. Following [8], we measure the relative informativeness $\mathbf{q}_i$ of a filter in ℓ -th layer by ℓ_1 -norm $\|\mathcal{F}_{\ell,i}\|_1$, and obtain

$$\mathbf{q} = [\|\mathcal{F}_{\ell,1}\|_1, \|\mathcal{F}_{\ell,2}\|_1, \dots, \|\mathcal{F}_{\ell,n_{\ell+1}}\|_1]^\top. \quad (5.4)$$

In Figure 5.3, we show the distribution of filters' ℓ_1 -norm for each convolutional layer in a C3D network well-trained on Sports1M dataset. The sum of all absolute weights of a filter provides a clue for the magnitude of the output feature maps. Compared with other filters in that layer, filters with larger weights are likely to produce feature maps with stronger activations. In this sense, our filter pruning method which preserves filters with large weights and is data-independent, is able to capture the information that is obtained from the activation-based feature maps pruning methods.

Filter Representation Φ_i . Motivated by the Maximum Activation of Convolutions (MAC) feature representations [51], which generates compact representations from feature maps of convolutional layers. The 3-dimensional feature maps consist of multiple channels, and MAC is obtained by taking the maximum value of each channel. Similar to feature maps, each filter can be seen as a pillar piled up by

FIGURE 5.4: Maximum Abs of Filters (MAF) representation. The filter cube with the maximal absolute values are concatenated as the compact filter representation Φ_i .

3-dimensional filter cube (see Figure 5.4). Every convolution operation with a filter cube will generate a numerical value in the corresponding feature map of next layer through sum of multiplications. Hence, only large weights have a chance to dominate the activations of the following feature maps. Based on this observation, we concatenate the weights with maximal absolute values of a filter cube, obtaining the compact filter representation $\Phi_i \in \mathbb{R}^{n_i}$. Then it is post-processed by ℓ_2 normalization. We term this kind of representations as MAF (Maximum Absolute value of Filters). The cosine similarity is utilized to measure pairwise filters:

$$S_{ij} = \Phi_i^\top \Phi_j. \quad (5.5)$$

5.2.3 Multiple Layers Pruning

Inspired by [8, 157], we prune a given network layer by layer. For each layer, we apply the k -DPPs sampling to all filters. Except for the selected k filters, the remaining will be pruned and the corresponding feature maps generated by those filters will consequently be removed, so will the filter cubes associated with those feature maps in the next convolutional layer (see Figure 5.5). Subsequently, the filters pruned in the ℓ -th layer can affect the number of feature maps generated by them and also the size of filters in $(\ell + 1)$ -th layer which operates convolution

FIGURE 5.5: Filter pruning across multiple layers. The filters pruned in the ℓ -th layer (green by column) will affect the number of feature maps generated by them and also the size of filters in $(\ell + 1)$ -th layer (green by row).

on previously removed feature maps. Besides, the filters in $(\ell + 1)$ -th layer will simultaneously be subject to pruning.

Since the learning capability of a network is determined by its architecture and the number of effective learnable parameters, removing a small number of unimportant filters will not affect the network accuracy significantly. However, a heavy pruning will inevitably degrade the accuracy. To compensate for the accuracy loss caused by pruning, we apply a retraining step after all layers have been pruned. Algorithm 2 summarizes our training process.

Algorithm 2 3D pruning for a well-trained network

Input: A well-trained network, and the pruning ratio $\rho \in (0, 1)$.

Output: Pruned network, the pruned neural network object including its weights.

- 1: **for** $\ell \leftarrow 1$ to L **do**
 - 2: Informativeness for each filter: $\mathbf{q}_i \leftarrow \|\mathcal{F}_{\ell,i}\|_1$
 - 3: Mutual similarity: $S_{ij} \leftarrow \Phi_i^\top \Phi_j$
 - 4: Build Objective from learned variable $\mathbf{q}_i$ and S_{ij}
 - 5: Optimize objective function using k-DPPs [30], get subset Y_ℓ
 - 6: **end for**
 - 7: **end for**
 - 8: Parameterize the network with $\{Y_\ell\}_{\ell=1}^L$
-

TABLE 5.1: Statistics of baseline models.

Network Dataset	C3D Sports1M	P3D Sports1M	VGG16		ResNet-18 CIFAR-10	ResNet-50		ResNet-101 ImageNet
			CIFAR-10	ImageNet		CIFAR-10	ImageNet	
#Params	8.0e+07	6.5e+07	1.5e+07		1.1e+07	2.6e+07		4.5e+07
#Flops	5.6e+09	1.2e+10	0.3e+09	1.5e+10	1.3e+09	3.2e+09	4.1e+09	7.8e+09
Top-1 Acc (%)	57.9	63.4	91.6	68.7	93.5	93.6	72.8	77.4
Top-5 Acc (%)	82.6	84.4	-	87.2	-	-	91.1	93.6

5.3 Experiments

To evaluate our proposed 3D ConvNets pruning method, we conduct experiments on the extensively used C3D network [158] and P3D network [159] for video classification on Sports-1M dataset [160]. Our proposed method can also be flexibly extended to 2D network pruning, which is evaluated on the popular VGG-16 network [161] and ResNet-18/50/101 [162] for image classification with datasets including ImageNet [18] and CIFAR-10 [163]. All experiments are performed on a 12G Tesla K40 GPU using PyTorch. The statistical model information and classification accuracy of the original model are shown in Table 5.1. The number of parameters is mainly dependent on the networks (minor difference on the last classification layer), while the number of Flops is related to both the width and depth of networks and also the input size. Although the same baseline network is used, the different input sizes of ImageNet ($224 \times 224 \times 3$) and CIFAR-10 ($32 \times 32 \times 3$) lead to the huge difference in the size of computations and also the model running time.

Since we did not know of any work specifically designed for 3D ConvNets pruning, we compare our 3D pruning method with some variations that are highly related to ours. As we apply DPPs sampling to select the informative and non-duplicate filters, we measure the informative score by the ℓ_1 -norm of each filter. We assume filters with larger ℓ_1 -norms contain more information. Considering our settings, the controlled experiments should be able to verify (1) whether large ℓ_1 -norm indicate high informativeness; and (2) whether the non-duplicate factor is useful. Based on this, we have made the following comparisons:

- **ℓ_1 -Max:** To evaluate the effectiveness of DPPs sampling, especially the non-duplicate factor, we compare with the setting that large ℓ_1 -norm is the only

criterion for selecting filters. The large ℓ_1 -norm policy is also applied in [8] for 2D filter pruning. Here, ℓ_1 -Max denotes the filters with larger ℓ_1 -norms are selected.

- **Random:** In [156], Sajid and Sung evaluate some random combinations of 2D filters, from which the combination that causes the least degradation to the network accuracy is selected. We compare our method with this random sampling strategy to validate our assumption that large ℓ_1 -norm indicates high importance.
- **ℓ_1 -Small:** For a comprehensive comparison, we also show the performance of pruned model on the setting that the filters with small ℓ_1 -norms are selected.

To save space, we abbreviate the above descriptions on comparisons as ℓ_1 -Max, Random and ℓ_1 -Small, respectively, in the following figures and tables.

5.3.1 3D ConvNets Pruning

Sports1M is a large video classification benchmark, consisting of about 1.1 million sports videos labeled as 487 sports categories. As sports1M has many long videos, we follow [158] to randomly extract five 2-second long clips from every video, and then sample 16 key frames from every clip. Clips are resized to have a frame size of 128×171 , and then centrally cropped into 112×112 . We report the video classification results. The details of the backbone networks are shown below:

1. **C3D.** C3D [158] is a 10-layer single-branch 3D convolutional neural network, with 8 convolutional layers. It is well-designed for capturing spatiotemporal features, and it has been widely used in video related topics, such as action recognition [164–166].
2. **P3D199.** In [159], multiple P3D variants are provided. We choose the complete design of P3D, dubbed P3D199, which is designed based on the

FIGURE 5.6: C3D (**Top**) and P3D (**Bottom**) network on Sports1M: Single layer performance analysis under different pruning ratios (w/o retraining).

152-layer ResNet [162]. It contains four stages of residual blocks, and the number of blocks in each stage is 3-8-36-3, respectively. Each block consists of four conv. layers, two vanilla 3D conv. layers located at the beginning and the end, between which are a spatial conv. layer and a temporal conv. layer. The pair of the spatial and temporal layers are integrated to construct the DPPs marginal matrix L for filters selection.

For the two backbone networks, we use the well-trained network on the Sports1M train split as our base network, and test the pruned model on the test split.

Single Layer Pruning. In this part, we compare our proposed 3D filter pruning method with the above mentioned baselines: ℓ_1 -Max, ℓ_1 -Random and ℓ_1 -Small in a layer-by-layer manner. Figure 5.6 shows the curves of classification accuracy on Sports1M changing with filter pruning ratios for all convolutional layers of C3D network and P3D199 network. As expected, the accuracy degrades as pruning ratio increasing. Our method and ℓ_1 -Max performs consistently better than ℓ_1 -Random and ℓ_1 -Small, which suggests that filters with larger (abs) values have better representative power. One can also observe that our 3D pruning method

FIGURE 5.7: Visualization of filters in conv1 of C3D network trained on Sports1M, with about 20% (12/64) pruning ratio. Filters are ranked by its importance score q_i in a descending order from top left to bottom right. Filters with red boxes are the ones pruned by our method. The last 12 filters circled by black dashed lines are the ones pruned according to criteria in [8], i.e., pruned the ones with least ℓ_1 -norm.

generally outperforms ℓ_1 -Max. The gaps between those two methods are due to the effective DPPs sampling catering for both representative and dicriminative filters. Interestingly, ℓ_1 -Random sometimes leads to better accuracy than ℓ_1 -Max. As we conjectured in Section 5.2.1, this may be caused by the duplicates among filters with large ℓ_1 -norm, i.e., the filters selected by ℓ_1 -Max strategy are not diverse enough.

To further validate this conjecture, we visualize the 3D filters in the first convolutional layer of C3D network in Figure 5.7. There are 64 filters in $\mathbb{R}^{3 \times 3 \times 3 \times 3}$ in total, and those filters are ranked by its ℓ_1 -norm in a descending order. If the ℓ_1 -Max strategy is applied to prune, say with ratio $= \frac{12}{64}$ (18.75%), then the top 52 filters will all be selected (filters except the ones circled by black dashed lines). It is obvious that some selected filters are highly similar to each other, and some pruned filters may contain unique information/knowledge. As a result, some functionally

TABLE 5.2: Comparison of several baselines and our proposed on Sports1M dataset. **Top: C3D. Bottom: P3D199.**

Pruning Ratio	Saved FLOPs	Max		Random		Small		Ours	
		Top-1(%)↓	Top-5(%)↓	Top-1(%)↓	Top-5(%)↓	Top-1(%)↓	Top-5(%)↓	Top-1(%)↓	Top-5(%)↓
10%	18.51%	0.22	0.13	0.42	0.26	1.21	0.47	0.1	0
20%	35.59%	0.78	0.35	1.7	0.74	2.59	0.79	0.36	0.11
30%	50.53%	1.41	0.91	2.52	1.62	4.07	2.63	0.93	0.44
40%	63.52%	2.76	1.84	4.94	2.77	9.73	6.57	1.76	0.89
50%	74.73%	5.11	3.32	8.4	4.39	14.82	10.83	3.79	1.94

Pruning Ratio	Saved FLOPs	Max		Random		Small		Ours	
		Top-1(%)↓	Top-5(%)↓	Top-1(%)↓	Top-5(%)↓	Top-1(%)↓	Top-5(%)↓	Top-1(%)↓	Top-5(%)↓
10%	18.48%	0.17	0.15	0.41	0.19	1.36	0.58	0.04	0
20%	35.08%	0.65	0.26	1.18	0.57	2.4	0.72	0.27	0.08
30%	49.79%	1.4	0.73	1.79	1.43	4.14	2.31	0.81	0.3
40%	62.61%	2.53	1.55	4.28	2.82	9.26	6.45	1.61	0.67
50%	73.56%	4.76	3.17	7.21	4.35	13.79	10.26	3.52	1.69

diverse filters are probably missed. To avoid such a circumstance, our DPPs-based method tends to select filters that are more representative while considering their similarities (filters except the ones circled by red boxes), resulting in a remarkable improvement over ℓ_1 -Max.

Multiple Layers Pruning. As introduced in Section 5.2.3, we prune the whole network layer by layer. That is, after filters being pruned in each layer, we combine all convolutional layers and trim the remaining filters according to the strategy showed in Figure 5.5. Then we fine-tune the pruned model with a decreasing learning rate starting from 0.0015 for 6 epochs on C3D and 0.0005 for 8 epochs on P3D199. The results are shown in Table 5.2. We achieve about 75% FLOPs reduction with only 1.94% Top-5 performance drop using C3D network, and 1.69% acc. drop using P3D network. The top-1 performance is still satisfactory at about 50% FLOPs reduction for both networks with less than 1% acc. drop. The P3D199, which is constructed based on Resnet-152, performs slightly better than C3D. One possible reason is that the residual design could preserve the information from previous layers that is pruned at current layer. It can be seen that the performances of ℓ_1 -Random and ℓ_1 -Small on both networks are undesirable, which again reflects the rationality of using ℓ_1 -norm as the indicator of importance for filters. The superior performance of our proposed method over ℓ_1 -Max demonstrates the effectiveness of using DPPs to select the representative and yet diverse filters.

TABLE 5.3: Comparison of SOTA and our proposed method on Sports1M dataset. **Top: C3D. Bottom: P3D199.**

Pruning Ratio	Saved FLOPs	Xavier <i>et al.</i> [91]		Luo <i>et al.</i> [93]		ℓ_1 -act.		Pavlo <i>et al.</i> [80]		Ours	
		Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓
10%	18.51%	0.52	0.33	0.13	0.02	1.32	0.51	0.11	0.04	0.10	0
20%	35.59%	1.35	0.74	0.40	0.15	2.24	0.86	0.40	0.14	0.36	0.11
30%	50.53%	1.72	1.15	0.90	0.45	3.97	2.54	0.97	0.54	0.93	0.44
40%	63.52%	3.35	2.19	1.91	0.93	7.39	5.44	1.59	0.88	1.76	0.89
50%	74.73%	5.28	3.96	4.02	2.12	10.28	7.65	4.00	2.03	3.79	1.94

Pruning Ratio	Saved FLOPs	Xavier <i>et al.</i> [91]		Luo <i>et al.</i> [93]		ℓ_1 -act.		Pavlo <i>et al.</i> [80]		Ours	
		Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓	Top-1(%) ↓	Top-5(%) ↓
10%	18.48%	0.47	0.30	0.07	0.02	1.41	0.63	0.05	0.01	0.04	0
20%	35.08%	1.22	0.83	0.32	0.14	2.32	0.91	0.26	0.11	0.27	0.08
30%	49.79%	1.85	1.29	0.97	0.40	4.10	2.67	0.91	0.57	0.81	0.30
40%	62.61%	3.46	2.27	1.88	0.75	9.45	6.14	1.60	0.62	1.61	0.67
50%	73.56%	6.08	4.19	3.96	1.53	10.47	8.65	3.60	1.47	3.52	1.69

Comparison to State-Of-The-Art (SOTA). There are mainly two directions in network pruning, weights based pruning and activation based pruning. The main difference of the two network pruning directions is that the former is data independent, so that given a well-trained network, we are able to directly remove the redundant weights. However, the activation based pruning compresses a network by measuring its activations, which are highly dependent on the inputs. And the pruned network needs to be updated once unseen samples are available. Thus the generalization ability of weights based pruning is superior to that of activation based, and the majority studies recently published are on weights pruning.

We compare our proposed method with both activation-based and weight-based methods [80, 91, 93, 167]. The first two compared methods, [93, 167] and [91], are representative research directions for activation based pruning. We use the released code from [93] and implement [91] according to their papers. We also design a baseline ℓ_1 -norm of a activations/feature maps, denoted as ℓ_1 -act. The basic idea of the third method [80] is to evaluate a filter by the performance drop after removing it from the network. In [80] a few variations are proposed. We use the first-order importance estimate as it is most frequently used in their experiments.

The results are shown in Table 5.3, from which we can see that ℓ_1 -act. performs notably worse than the other methods. Considering that ℓ_1 -norm of filters is the prevailing measurement for weights based methods, the observation suggests that the importance of a feature map is not closely related to the ℓ_1 -norm as to filters. Although better than ℓ_1 -act., [91] is still inferior to our method by a large margin. ThiNet [93] is comparable to our method, however, its computational complexity is $\mathcal{O}(C_{n_\ell}^k)$, while ours is $\mathcal{O}(n_\ell k^3)$. Our method is more efficient at $k < n_\ell$, and the gap enlarges with the decrease of k . The weight-based method [80] performs slightly worse than ours in most cases. The possible reason is that [80] is under an *i.i.d.* assumption of all filters, and the importance of each filter is evaluated individually based on its influence to the ultimate classification accuracy. However, we are of the view that the relations between filters matter. A large set of informative but duplicate filters is sub-optimal and potentially subject to further

TABLE 5.4: Comparisons of the proposed MAF to the naive vectorized filter representation (Top-1 Acc drop (%), with fine-tuning).

Pruning Ratio	C3D/Sports1M MAF / Vec.	P3D/Sports1M MAF / Vec.	VGG16/CIFAR-10 MAF / Vec.	VGG16/ImageNet MAF / Vec.
10%	0.1 / 0.13	0.04 / 0.04	0.0 / 0.0	0.80 / 0.71
20%	0.36 / 0.40	0.27 / 0.26	0.06 / 0.04	1.24 / 1.25
30%	0.93 / 0.93	0.81 / 0.83	0.14 / 0.15	1.97 / 1.94
40%	1.76 / 1.81	1.61 / 1.70	0.29 / 0.26	2.63 / 2.70
50%	3.76 / 3.78	3.52 / 3.55	0.41 / 0.40	3.83 / 3.76

pruning into a small set of representative ones, or being improved by the same number of informative yet diverse filters, which is the key difference between our method and [80].

Ablation Study. To evaluate the effectiveness of the proposed compact filter representation Maximum Abs. of Filters (MAF), we conduct experiments using the naive vectorized filter representation for an ablation study. As our method is weights based pruning, which is closely related to the activation based pruning, we thus compare with several representative works on activation based methods.

We compare the proposed MAF with the naive vectorized filter representation (abbreviated as Vec.) while applying to the filter pruning for the whole network. Table 5.4 shows the results. From the table we can see that the MAF is slightly superior to Vec. filter on 3D network pruning, and is comparable to Vec. filter on 2D network pruning. One possible reason is that the redundancy in 3D network is more serious than that in 2D network. Besides, the proposed MAF is much more compact than the vectorized filter representation, so that it consumes less memory for storage and short time for similarity calculation.

5.3.2 2D ConvNets Pruning

For a fair comparison with existing filter pruning method, we extend our proposed 3D pruning method to 2D ConvNets. The importance score $\mathbf{q}_i$ is calculated similar to 3D ConvNets using ℓ_1 -norm. The filter representation Φ_i is now obtained by

FIGURE 5.8: VGG-16 network on CIFAR-10 (**Top**) and ImageNet (**Bottom**): Single layer performance analysis with different pruning ratios (w/o retraining).

taking the max (abs) value of 2-dimensional kernels rather than the filter cube in 3D ConvNets. Cosine similarity is applied to measure the relationships between filters. We evaluate the extensively used VGG-16 network and ResNet-18/50/101 on CIFAR-10 and ImageNet.

Single Layer Pruning. Figure 5.8 shows several representative layers' pruning results of VGG-16 on CIFAR-10 and ImageNet. On both datasets, our proposed method and ℓ_1 -Max perform marginally better than ℓ_1 -Random and ℓ_1 -Small, which is consistent with our observation on 3D ConvNets. For VGG-16 on CIFAR-10, the results vary a lot in different layers. Specifically, the filter pruning on last layers does not influence the classification results due to the small size of input image and the relative very deep network architecture. For VGG-16 on ImageNet, the accuracy drops dramatically with pruning ratio increasing, which suggests the majority of filters are critical for a satisfactory performance.

Comparison to SOTA w.r.t. Multiple Layers Pruning. The networks are pruned layer by layer, and then they are fine-tuned to boost the performance. For VGG-16 on CIFAR-10/ImageNet, it is fine-tuned for 10/23 epochs with a fixed learning rate 0.001. For ResNet-18/-50 on CIFAR-10, it is fine-tuned for 10/12 epochs with a decreasing learning rate from 0.01. For ResNet-50/-101 on ImageNet, it is fine-tuned for 25/30 epochs with a decreasing learning rate from 0.01. Table 5.5 and Table 5.6 show comparisons of various networks on CIFAR-10

TABLE 5.5: Comparisons of pruning results on Cifar10 in terms of various pruning ratios (PR).

Network	Method	PR	FLOPs ↓	Acc.(%) ↓
VGG16	Li <i>et al.</i> [8]	-	34.2%	0.13
	Huang <i>et al.</i> [168]	-	55.2%	1.1
	Hu <i>et al.</i> [169]	-	56.2%	0.24
	Ours	10%	18.29%	0.0
		20%	35.27%	0.06
		30%	50.45%	0.14
		40%	63.45%	0.29
		50%	74.86%	0.41
		60%	83.58%	0.97
		70%	90.65%	1.72
		80%	95.79%	2.26
ResNet-18	Li <i>et al.</i> [8]	7.2%	7.5%	1.06
	Huang <i>et al.</i> [168]	-	64.5%	1.9
	Pavlo <i>et al.</i> [80]	-	25.0%	0.44
	Yang <i>et al.</i> [79]	40%	54.0%	1.76
	Ours	10%	11.25%	0.0
		20%	22.93%	0.05
		30%	33.96%	0.11
		40%	43.71%	0.24
		50%	54.06%	0.36
ResNet-50	Li <i>et al.</i> [8]	13.7%	27.6%	1.73
	Hu <i>et al.</i> [169]	20%	56.2%	0.49
	Pavlo <i>et al.</i> [80]	-	25.0%	0.32
	Yang <i>et al.</i> [79]	40%	52.6%	0.1
	Lin <i>et al.</i> [84]	-	48.5%	0.52
	Xavier <i>et al.</i> [91]	-	61.5%	0.40
	Ours	10%	13.13%	0.0
		20%	23.77%	0.04
		30%	36.21%	0.09
		40%	45.26%	0.20
		50%	55.14%	0.31

and ImageNet. For VGG-16 on CIFAR-10, 34.2% FLOPs reduction is achieved with 0.13% accuracy drop in [8] and 55.2% FLOPs reduction with 1.1% accuracy drop in [168]. Our proposed method outperforms [8, 168] by 0.06% accuracy drop at 35.27% FLOPs saving and 0.29% accuracy drop at 63.45% FLOPs saving, respectively. [169] and our method performs similarly with 0.24% acc. drop at 56.2% FLOPs saving versus 0.29% acc. drop at 63.45% FLOPs saving. For the ResNet, the pruning ratio of [8] is clearly lower than our method, while [168] and [169] are promising on saving FLOPs, but suffer from relatively large accuracy drop. For those recent work from CVPR'19 like [79, 80, 84, 91], they usually save less FLOPs

TABLE 5.6: Comparisons of pruning results on ImageNet in terms of various pruning ratios (PR).

Network	Method	PR	FLOPs↓	Top-1(%)↓	Top-5(%)↓
VGG16	Pavlo [170]	-	72.9%	5.12	5.44
	Li [8]	-	15.5%	0.76	0.51
	Luo [93]	-	68.39%	2.98	1.5
	Lin [171]	-	58.71%	1.52	0.65
	Ours	10%	18.83%	0.8	0.44
		20%	35.52%	1.24	0.7
		30%	50.71%	1.97	1.12
		40%	63.57%	2.63	1.78
		50%	74.90%	3.83	2.42
ResNet-50	Li [8]	30%	36.78%	2.56	1.49
	Luo [93]	30%	36.78%	0.84	0.47
	Lin [171]	30%	36.78%	3.09	1.63
	Yang [79]	40%	53.5%	1.32	0.55
	Lin [84]	16.8%	43.02%	4.20	1.93
	Lin [84]	42.5%	61.36%	6.27	3.12
	Zhao [85]	40%	-	2.30	1.70
	Ours	10%	13.45%	0.07	0.02
		20%	24.61%	0.37	0.16
		30%	36.78%	0.94	0.40
		40%	45.73%	1.73	1.12
		50%	55.82%	2.61	1.39
ResNet-101	Li [8]	30%	39.62%	0.33	0.24
	Li [8]	50%	65.14%	1.24	0.71
	Pavlo [80]	30%	39.74%	0.02	-
	Yang [79]	30%	42.2%	0.05	0.0
	Ours	10%	14.19%	0.01	0.0
		20%	27.30%	0.06	0.02
		30%	41.52%	0.11	0.07
		40%	53.67%	0.25	0.10
		50%	72.49%	0.52	0.21

under the same rate of accuracy drop, except the case of [79] on ResNet-50, where the accuracy drop is slightly better by 0.1%.

We can observe similar phenomenons on larger datasets and deeper networks. For VGG-16 on ImageNet, although [171] is slightly better than ours, the method requires extra prior knowledge about the importance of filters which leads to an unfair comparison. The proposed method performs favourably against [8] and [170], and is comparable to [93] with 2.63% acc. drop at 63.57% FLOPs saving verses 2.98% acc. drop at 68.39% FLOPs saving. Note that [8] is only based on ℓ_1 -norm for selecting filters, i.e., the baseline of ℓ_1 -Max, the inferior performance

on 2D ConvNet pruning again verifies the effectiveness of our proposed DPPs-based method. For ResNet-10/101 on ImageNet, [8], [93] and [171] achieve lower performances than our method if we consider pruning ratio, FLOPs and accuracy drop in a whole. For those recent work from CVPR'19, [80] and [79] demonstrate promising performances on par with our method, while [84] and [85] suffer from significant accuracy drop, i.e., 6.27% and 2.30%, respectively.

5.4 Conclusion

In this chapter, we have proposed a new filter pruning method for accelerating 3D ConvNets. Filter pruning for 3D ConvNets is critical to deal with large volume of video data and it is feasible owing to the considerable redundancy in consecutive video frames. To our best knowledge, the proposed method is a pioneer work to explore filter pruning in 3D ConvNets. We use DPPs, which integrate filter importance and the similarities between filters into a unified model, for selecting informative yet diverse 3D filters. Additionally, a novel filter representation MAF is introduced to measure the similarities between filters. Our proposed method achieves 50.53% FLOPs reduction with only 0.93% drop in accuracy for C3D network on Sports1M dataset. We also extend our method to 2D filter pruning and achieve performance comparable with state-of-the-art filter pruning methods for 2D ConvNets.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis studies the common pattern discovery and its combination with CNN models, and shows that the CNN strategies could strongly enhance the performance of common pattern discovery, which in turn helps to address problems in CNNs. Several task-driven methods are proposed in this thesis and quantitatively evaluated by applying to computer vision tasks, including object localization, image and instance search, noisy image classification and action recognition from videos.

First, we propose a framework for unsupervised common object discovery and apply it to the task of object localization from wild images and also image/instance retrieval. Comparing to existing methods, we tackle a more complex scenario where an image could contain multiple common objects instead of at most one as in the literature, and the proportion of images without common objects could be as high as 60% compared to 15% in the literature. Our proposed method takes advantage of pretrained CNNs to extract visual primitives and also the feature representations. Due to lack of annotation, we formulate the unsupervised object discovery as a sub-graph mining problem where the visual primitives extracting from images are nodes of graphs and their similarities are edges. An efficient optimization algorithm

is proposed to solve it. We conduct extensive experiments on the Object Discovery dataset, and PASCAL VOC 2007 dataset for object localization, and on Holidays dataset and Oxford5k dataset for image search. The results show that the proposed method performs well in a fully unsupervised way, and even when variations in scale, view-point, appearance or partial occlusions frequently occur in images.

Second, observing the heavy relying of CNN models on images with high-quality annotations, we propose to use the images directly collected from websites to train the CNN-based classifiers so as to reduce the cost of manual annotation. The main challenge is that the labels of these images are usually coarse or inaccurate. Previous methods address this noisy label/image learning problem either by improving the image label quality or building robust models. We combine both strategies and propose to purify the noisy images first by evaluating their similarities to common patterns of each class and then feed the relabeled images to our noisy-robust CNN model. We evaluate the proposed methods on both synthesized label noise datasets and real-world cloth dataset which contains noisy samples, and achieve competitive performance in the noisy learning field.

Third, by treating the filters in CNN models as visual primitives, we propose a novel filter pruning method to remove the redundant filters so as to accelerate the over-parametrized 3D CNN models. Previous works center on pruning 2D CNNs; however, the 3D CNNs being designed for video processing contain more redundancy and the speed for video processing is more critical. Our work bridges this gap and achieves promising performance on the widely-used 3D CNNs, such as C3D and P3D in the task of action recognition. For example, the accuracy only drops 0.93% while reducing 50.53% computations. In addition, the proposed method could be extended to 2D CNN models pruning and also achieves among state-of-the-art performance.

6.2 Future Work

This thesis centers on the intersection of common pattern discovery and deep learning, i.e., CNN models. Three research directions can be explored as future work.

6.2.1 CNN Pruning from Feature Maps

We have considered filter-based CNN pruning in Chapter 5, which treats filters as visual primitives. The other widely-used method is to identify the redundant feature maps and prune away the corresponding filters that generate them. As the redundant filters and feature maps are not always consistent, there is some difference between the two methods. The filter-based method optimizes CNNs based on their weights without considering the specific input domain; thus it has better generalization ability; while feature-maps-based method optimizes CNNs based on the outputs of each layer, i.e., feature maps. The results vary along different inputs, but the performance is better if the testing data domain is similar to the training data domain. Thus, the feature maps based method is also worth studying.

As seen in Figure 6.1, the feature maps outputting from layers also show patterns, which are still obvious after three convolutional layers. Thus it is straightforward to apply common pattern discovery strategy to feature maps based pruning. One possible solution is to follow the filter based pruning, the individual informativeness

FIGURE 6.1: Visualization of feature maps output from a CNN layer (Left) and the corresponding patches in the original images (Right).

and similarities can be used for selecting the important and diverse feature maps, so the corresponding filters that output them are retained. The remaining redundant feature maps are to be removed by pruning the filters that generate them.

The key of this approach is to define informativeness of an individual feature map and their similarities. There are several ways to measure the similarity between two features, such as Euclidean distance. However, few works are on evaluating the importance of the feature maps. Hence, this future work could start with analyzing the critical characteristics of a useful feature map and further apply it to accelerate the over-parameterized CNNs.

6.2.2 Video Summarization

Automatically summarizing videos to short clips or key frames are useful, such as movie trailers, so that people can have a quick view of a long video. However, there is no standard examples or ground-truth summarizations to a long video as people may be attracted by different plots. Following our work in Chapter 3 which is to mine the common objects from images, we could also mine the frequently appeared objects/foregrounds from videos, then the resulting objects and the frames containing these objects could be representatives of the videos. Except for such a general summarization, it is also possible to customize the video summarizations by integrating user preference to object candidatures.

Specifically, we could extract visual primitives from frames by CNN-based object proposal methods. Based on observation that the leading characters/objects of a video appear more frequently, we can discover the high-frequency objects from the visual primitives pool. Then the ones containing these objects are selected frames to represent the video. Compared to common object discovery from images, it is more challenging to explore videos as (1) the number of visual primitives and the number of object classes are both much larger than that in images; (2) the appearance of objects varies a lot in different frames, making it difficult to find the class patterns. Thus, this work requires more efforts on designing strategies to

FIGURE 6.2: Examples of diverse image retrieval.

extract high quality visual primitives, and also on building models that can better capture the features of varying objects in videos.

6.2.3 Diverse Image Retrieval

Imaging you are shopping online, the search engine automatically recommends products to you based on your search history; which kind of products you would like to be recommended? A list of exactly same products or a list of similar but diverse products (see Figure 6.2)? Traditional image retrieval methods are evaluated only by the similarity between the reference images and the query, without considering the relations among the reference images. However, in practical usage, such as recommendation system for shopping website, users would not like to see the same product repeating ten times on their screen. To address this problem, we aim to design a model that could recognize the main features of a product of interest, and recommend customers with products containing the common features while being different to one another.

Dataset. Since there is no such a dataset for diverse image retrieval, we can explore this problem by using image sets for fine-grained classification, including CUB bird dataset [172] and Oxford Flower dataset [173]. The fine-grained classification is to distinguish subcategories, so that the images have common features as their supercategory and are also slightly different across subcategories, which naturally fit the scenario of diverse image retrieval. The bird dataset consists of bird images

in 10 subcategories, and it is the same case for the flower dataset. The two datasets can be combined with 10 bird classes being labeled as $0 \sim 9$, and the flower classes as $10 \sim 19$. Then the concrete goal of this setting is to find the images with the same category as query image, but in different subcategories. For example, given a query of bird, our ideal top-10 retrieval results are the images from 10 different bird subcategories, rather than that from the same subcategory as query or flower images.

Problem Formulation. Given a query image I_q and a set of reference images $\mathcal{Y} = \{I_1, I_2, \dots, I_N\}$, our goal is to select a subset of images $Y \subseteq \mathcal{Y}$, $|Y| = k$, that are similar to the query but also slightly different to one another. Here N denotes the total number of reference images, and k is set as the number of selected images. Motivated by the successful application of Determinantal Point Processes (DPPs) to key frames selection from videos, where the goal is to select important frames that are not duplicate,s we utilize DPPs method [101] to solve our diverse image retrieval problem. It can be formulated as:

$$Y = \arg \max_s \frac{\det(L_{Y_s})}{\sum_{t=1}^{\mathcal{C}_N^k} \det(L_{Y_t})}, \quad (6.1)$$

where Y_s is the s -th subset, and $L \in \mathbb{R}^{N \times N}$ is a semidefinite matrix with each element being defined as $L_{ij} = \mathbf{q}_i \Phi_i^\top \Phi_j \mathbf{q}_j$. Here $\mathbf{q}_i, \mathbf{q}_j$ are values depicting the similarities of i -th, j -th images to the query I_q , and Φ_i, Φ_j are feature vectors of reference images. L_{Y_s} is calculated from the candidate reference in subset Y_s .

Note that the number of candidature subset could be as large as $\mathcal{C}_N^k$. To reduce computations, we only search from images that are top- T neighbors to the query instead of all reference images, and the neighbors are defined based on similarities to the query, i.e., $\langle I_i, I_q \rangle = \Phi_i^\top \Phi_q$. Then the number of candidature subset is significantly reduced to $\mathcal{C}_T^k$.

Experiments. We use mean Average Precision (mAP), which is a popular metric used to measure the performance of models doing retrieval, to measure our diverse retrieval results. Table 6.1 lists the results vary with k and T , respectively denoting

TABLE 6.1: Experimental results (mAP (%)) of our diverse image retrieval.

Our Method	k=10	k=20	k=30	k=40	k=50	k=60	k=70	k=80	k=90	k=100
T=200	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
T=300	0.98	0.98	0.98	0.98	0.97	0.97	0.98	0.97	0.97	0.97
T=400	0.95	0.95	0.94	0.94	0.93	0.94	0.92	0.92	0.93	0.92
T=500	0.89	0.91	0.90	0.89	0.88	0.86	0.86	0.87	0.85	0.86
T=N	0.73	0.70	0.72	0.72	0.70	0.70	0.7	0.69	0.69	0.7

the size of subset and the number of neighbor images used. The column-wise results drop with increasing T , showing that the step to select neighbors is critical to our proposed method; and the row-wise results are stable with increasing k , meaning that our method is robust to the size of subset once the number of reference images is fixed.

Discussion. We have conducted a toy experiment on the problem of diverse image retrieval, and many problems are still unsolved. We list a few:

1. *Huge computational complexity.* The principal of our method is to enumerate all the subsets and score them by Equation (6.1). The number of subsets increases exponentially with the number of reference images, and the calculation of determinant, i.e., $\det(L_{Y_s})$, is at $\mathcal{O}(k^3)$. The huge computational cost is prohibitive for practical usage.
2. *Lack of Benchmarks.* As we mentioned, there is no such an image dataset for studying the topic. Manually collecting and annotating a large scale dataset is expensive.
3. *Lack of Baselines.* To fairly show the performance of a newly proposed method, there are usually several baselines to compared with. However, except the traditional retrieval method, we have not figured out more appropriate methods as baselines.

To summarize, there are many works can be extended from this thesis. In the near future, we shall explore the common pattern discovery both in theory and applications.

List of Author's Publications

1. **Zhenzhen Wang**, Chunyan Xu, Yap-Peng Tan and Junsong Yuan, Attention-Aware Label Noise Learning for Image Classification, arXiv preprint arXiv:2009.14757 (2020).
2. **Zhenzhen Wang**, Weixiang Hong, Yap-Peng Tan and Junsong Yuan, Pruning 3D Filters for Accelerating 3D ConvNets, IEEE Trans. on Multimedia (TMM), 2019.
3. Jiaqi Zhang, **Zhenzhen Wang**, Jingjing Meng, Yap-Peng Tan, and Junsong Yuan, Boosting Positive and Unlabeled Learning for Anomaly Detection with Multi-features, IEEE Trans. on Multimedia (TMM), 2018.
4. **Zhenzhen Wang** and Junsong Yuan, Simultaneously Discovering and Localizing Common Objects in Wild Images, IEEE Trans. on Image Processing (TIP), 2018.
5. Weixiang Hong, **Zhenzhen Wang**, Ming Yang, and Junsong Yuan, Conditional Generative Adversarial Network for Structured Domain Adaptation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
6. **Zhenzhen Wang** and Junsong Yuan, Unsupervised Multiple-Instance Learning for Instance Search, IEEE International Conference on Multimedia and Expo (ICME), 2018.
7. **Zhenzhen Wang**, Jingjing Meng, Tan Yu, and Junsong Yuan, Common Visual Pattern Discovery and Search, IEEE Conference APSIPA ASC, 2017.

8. Tan Yu, **Zhenzhen Wang**, and Junsong Yuan, Compressive Quantization for Fast Object Instance Search in Videos, IEEE International Conference on Computer Vision (ICCV), 2017
9. Jiaqi Zhang, **Zhenzhen Wang**, Junsong Yuan, and Yap-Peng Tan, Positive and Unlabeled (PU) Learning for Anomaly Detection with Multi-features, ACM International Conference on Multimedia (MM), 2017.

Bibliography

- [1] Zhenzhen Wang, Jingjing Meng, Tan Yu, and Junsong Yuan. Common visual pattern discovery and search. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1011–1018. IEEE, 2017. [iii](#)
- [2] Zhenzhen Wang and Junsong Yuan. Simultaneously discovering and localizing common objects in wild images. *IEEE Transactions on Image Processing*, 27(9):4503–4515, 2018. [iii](#)
- [3] Zhenzhen Wang and Junsong Yuan. Unsupervised multiple-instance learning for instance search. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2018. [iii](#)
- [4] Zhenzhen Wang, Chunyan Xu, Yap-Peng Tan, and Junsong Yuan. Attention-aware noisy label learning for image classification. *arXiv preprint arXiv:2009.14757*, 2020. [iv](#)
- [5] Zhenzhen Wang, Weixiang Hong, Yap-Peng Tan, and Junsong Yuan. Pruning 3d filters for accelerating 3d convnets. *IEEE Transactions on Multimedia*, 2019. [iv](#)
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. [xii](#), [6](#), [34](#), [53](#)
- [7] Weicheng Kuo, Bharath Hariharan, and Jitendra Malik. Deepbox: Learning objectness with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2479–2487, 2015. [xii](#), [4](#), [26](#), [34](#), [36](#), [39](#), [43](#), [44](#)
- [8] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *International Conference on Learning Representations (ICLR)*, 2017. [xiv](#), [74](#), [77](#), [78](#), [79](#), [80](#), [83](#), [85](#), [91](#), [92](#), [93](#)
- [9] Junsong Yuan. Discovering visual patterns in image and video data: Concepts, algorithms, experiments. Saarbrücken, Germany: VDM Verlag Dr. Müller, 2011. [1](#)

- [10] Tinne Tuytelaars, Christoph H Lampert, Matthew B Blaschko, and Wray Buntine. Unsupervised object discovery: A comparison. *International journal of computer vision*, 88(2):284–302, 2010. [1](#)
- [11] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. [3](#)
- [12] Congcong Li, Devi Parikh, and Tsuhan Chen. Automatic discovery of groups of objects for scene understanding. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2735–2742. IEEE, 2012. [3](#)
- [13] Tinne Tuytelaars, Krystian Mikolajczyk, et al. Local invariant feature detectors: a survey. *Foundations and trends® in computer graphics and vision*, 3(3):177–280, 2008. [4](#), [14](#)
- [14] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. [4](#), [14](#)
- [15] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010. [4](#), [14](#)
- [16] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. [4](#), [14](#), [34](#), [36](#), [39](#), [43](#), [44](#)
- [17] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3286–3293, 2014. [4](#), [14](#)
- [18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference Computer Vision and Pattern Recognition (CVPR)*, 2009. [5](#), [82](#)
- [19] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold mil training for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2409–2416, 2014. [13](#), [15](#), [24](#), [36](#), [42](#)
- [20] Xinggang Wang, Zhuotun Zhu, Cong Yao, and Xiang Bai. Relaxed multiple-instance svm with application to object discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1224–1232, 2015. [13](#), [15](#)

- [21] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018. [13](#)
- [22] Michael Rubinstein, Armand Joulin, Johannes Kopf, and Ce Liu. Unsupervised joint object discovery and segmentation in internet images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1939–1946, 2013. [15](#), [35](#), [36](#), [37](#), [38](#)
- [23] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic correlation promoted shape-variant context for segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8885–8894, 2019. [13](#)
- [24] Zhenzhen Wang, Xiao-Tong Yuan, Qingshan Liu, and Shuicheng Yan. Additive nearest neighbor feature maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2866–2874, 2015. [14](#)
- [25] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really? *arXiv preprint arXiv:1406.6962*, 2014. [14](#)
- [26] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2016. [14](#)
- [27] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. [14](#)
- [28] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. Prime object proposals with randomized prim’s algorithm. In *Proceedings of the IEEE international conference on computer vision*, pages 2536–2543, 2013. [14](#)
- [29] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998. [14](#)
- [30] Eugene L Lawler. *Combinatorial optimization: networks and matroids*. Courier Corporation, 2001.
- [31] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Mining and-or graphs for graph matching and object discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 55–63, 2015. [14](#)
- [32] Leonard Kaufman and Peter Rousseeuw. *Clustering by means of medoids*. North-Holland, 1987. [14](#)
- [33] Tan Yu, Yuwei Wu, Sreyasee Das Bhattacharjee, and Junsong Yuan. Efficient object instance search using fuzzy objects matching. In *AAAI*, pages 4320–4326, 2017. [14](#), [16](#), [48](#)

- [34] Tal Hassner and Ce Liu. *Dense Image Correspondences for Computer Vision*. Springer, 2016. [14](#)
- [35] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1511–1519, 2015. [15](#)
- [36] Zhiyuan Shi, Timothy M Hospedales, and Tao Xiang. Bayesian joint modelling for object localisation in weakly labelled images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):1959–1972, 2015. [15](#), [24](#), [25](#), [36](#), [42](#)
- [37] Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Semantic segmentation with context encoding and multi-path decoding. *IEEE Transactions on Image Processing*, 29:3520–3533, 2020. [15](#)
- [38] Alon Faktor and Michal Irani. Co-segmentation by composition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1297–1304, 2013. [15](#)
- [39] Gangqiang Zhao, Junsong Yuan, and Gang Hua. Topical video object discovery from key frames by modeling word co-occurrence prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1602–1609, 2013. [15](#)
- [40] David Liu, Gang Hua, and Tsuhan Chen. A hierarchical visual model for video object summarization. *IEEE Transactions on Pattern Analysis and Machine intelligence*, 32(12):2178–2190, 2010.
- [41] Cewu Lu, Renjie Liao, and Jiaya Jia. Personal object discovery in first-person videos. *IEEE Transactions on Image Processing*, 24(12):5789–5799, 2015. [15](#)
- [42] Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *Proceedings of the European Conference on Computer Vision*, pages 73–86. Springer, 2012. [15](#)
- [43] Yong Jae Lee, Alexei A Efros, and Martial Hebert. Style-aware mid-level representation for discovering visual connections in space and time. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1857–1864, 2013. [15](#)
- [44] Kevin Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1464–1471, 2014. [15](#), [25](#), [28](#), [32](#), [34](#), [36](#), [38](#), [41](#)
- [45] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proceedings of the IEEE Conference on Computer*

- Vision and Pattern Recognition*, pages 1201–1210, 2015. [28](#), [36](#), [38](#), [40](#), [41](#), [42](#), [43](#)
- [46] Armand Joulin, Kevin D Tang, and Fei-Fei Li. Efficient image and video co-localization with frank-wolfe algorithm. In *Proceedings of the European Conference on Computer Vision*, pages 253–268. Springer, 2014. [15](#), [25](#), [34](#), [36](#), [42](#), [43](#)
- [47] Alon Faktor and Michal Irani. “clustering by composition”—unsupervised discovery of image categories. In *Proceedings of the European Conference on Computer Vision*, pages 474–487. Springer, 2012. [16](#)
- [48] Armand Joulin, Francis Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1943–1950. IEEE, 2010. [16](#), [36](#), [38](#)
- [49] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 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, 2014. [16](#), [47](#)
- [50] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016. [16](#), [47](#), [48](#)
- [51] Giorgos Tolias, Ronan Sifre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *Proceedings of the International Conference on Learning Representations*, 2016. [16](#), [46](#), [48](#), [79](#)
- [52] Ashish Kankaria. Query expansion techniques. *Indian Institute of Technology Bombay, Mumbai, CSI Journal of Computing*, 1(2), 2012. [16](#)
- [53] Nikolaj Tatti and Aristides Gionis. Density-friendly graph decomposition. In *Proceedings of the International Conference on World Wide Web*, pages 1089–1099. International World Wide Web Conferences Steering Committee, 2015. [16](#)
- [54] Aristides Gionis, Flavio Junqueira, Vincent Leroy, Marco Serafini, and Ingmar Weber. Piggybacking on social networks. *Proceedings of the Very Large Data Base Endowment*, 6(6):409–420, 2013. [16](#)
- [55] Gangqiang Zhao and Junsong Yuan. Discovering thematic patterns in videos via cohesive sub-graph mining. In *Proceedings of the IEEE International Conference on Data Mining*, pages 1260–1265. IEEE, 2011. [16](#)
- [56] Gangqiang Zhao, Junsong Yuan, Jiang Xu, and Ying Wu. Discovering the thematic object in commercial videos. *IEEE Transactions on MultiMedia*, 18(3):56–65, 2011. [16](#)

- [57] Peng Cui, Shaowei Liu, and Wenwu Zhu. General knowledge embedded image representation learning. *IEEE Transactions on Multimedia*, 20(1):198–207, 2018. [18](#)
- [58] Kyunghyun Cho, Aaron Courville, and Yoshua Bengio. Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886, 2015. [18](#)
- [59] Andrea Malossini, Enrico Blanzieri, and Raymond T Ng. Detecting potential labeling errors in microarrays by data perturbation. *Bioinformatics*, 22(17):2114–2121, 2006. [18](#)
- [60] Jiang-wen Sun, Feng-ying Zhao, Chong-jun Wang, and Shi-fu Chen. Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007)*, volume 1, pages 244–250. IEEE, 2007. [18](#)
- [61] Dragan Gamberger, Nada Lavrač, and Sašo Džeroski. Noise elimination in inductive concept learning: A case study in medical diagnosis. In *Algorithmic Learning Theory*, pages 199–212. Springer, 1996.
- [62] Piyasak Jeatrakul, Kok Wai Wong, and Chun Che Fung. Data cleaning for classification using misclassification analysis. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 14(3):297–302, 2010. [18](#)
- [63] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2018. [18](#), [60](#), [63](#), [68](#), [69](#)
- [64] Jakramate Bootkrajang and Ata Kabán. Label-noise robust logistic regression and its applications. In *Machine Learning and Knowledge Discovery in Databases*, pages 143–158. Springer, 2012. [19](#)
- [65] Jakramate Bootkrajang and Ata Kabán. Boosting in the presence of label noise. *arXiv preprint arXiv:1309.6818*, 2013. [19](#)
- [66] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. [19](#)
- [67] Yunna Duan, Ying Gao, Xiaojuan Ren, Haoyuan Che, and Keyang Yang. Semi-supervised classification and noise detection. In *Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on*, volume 1, pages 277–280. IEEE, 2009. [19](#)
- [68] Dinesh Jayaraman and Kristen Grauman. Zero-shot recognition with unreliable attributes. In *Advances in Neural Information Processing Systems*, pages 3464–3472, 2014. [19](#)

- [69] Junbiao Pang, Fei Tao, Chunjie Zhang, Weigang Zhang, Qingming Huang, and Baocai Yin. Robust latent poisson deconvolution from multiple features for web topic detection. *IEEE Transactions on Multimedia*, 18(12):2482–2493, 2016. [19](#)
- [70] Sainbayar Sukhbaatar and Rob Fergus. Learning from noisy labels with deep neural networks. *International Conference on Learning Representations (ICLR)*, 2015. [19](#), [20](#), [56](#), [58](#), [62](#), [68](#), [71](#), [72](#)
- [71] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. [19](#), [51](#), [60](#), [63](#), [68](#), [69](#), [71](#), [72](#)
- [72] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pages 1–9, 2017. [20](#), [56](#)
- [73] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010. [20](#)
- [74] Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *Proceedings of the twenty-first international conference on Machine learning*, page 18. ACM, 2004. [20](#)
- [75] Jiaqi Zhang, Zhenzhen Wang, Jingjing Meng, Yap-Peng Tan, and Junsong Yuan. Boosting positive and unlabeled learning for anomaly detection with multi-features. *IEEE Transactions on Multimedia*, 2018.
- [76] Song Bai, Xiang Bai, and Wenyu Liu. Multiple stage residual model for image classification and vector compression. *IEEE Transactions on Multimedia*, 18(7):1351–1362, 2016. [20](#)
- [77] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006. [20](#), [58](#), [60](#)
- [78] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. [20](#), [58](#), [59](#)
- [79] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [21](#), [91](#), [92](#), [93](#)
- [80] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [21](#), [87](#), [88](#), [89](#), [91](#), [92](#), [93](#)

- [81] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017. [21](#)
- [82] Vadim Lebedev and Victor Lempitsky. Fast convnets using group-wise brain damage. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [21](#), [77](#), [78](#)
- [83] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems (ANIPS)*, 2016. [21](#), [77](#), [78](#)
- [84] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann. Towards optimal structured cnn pruning via generative adversarial learning. *arXiv preprint arXiv:1903.09291*, 2019. [21](#), [91](#), [92](#), [93](#)
- [85] Chenglong Zhao, Bingbing Ni, Jian Zhang, Qiwei Zhao, Wenjun Zhang, and Qi Tian. Variational convolutional neural network pruning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [21](#), [92](#), [93](#)
- [86] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *International Conference on Computer Vision (ICCV)*, 2017. [21](#)
- [87] Shaohui Lin, Rongrong Ji, Yuchao Li, Cheng Deng, and Xuelong Li. Toward compact convnets via structure-sparsity regularized filter pruning. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 2019. [21](#)
- [88] Yuhui Xu, Yongzhuang Wang, Aojun Zhou, Weiyao Lin, and Hongkai Xiong. Deep neural network compression with single and multiple level quantization. In *AAAI Conference on Artificial Intelligence*, 2018. [21](#)
- [89] Jianbo Ye, Xin Lu, Zhe Lin, and James Z Wang. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. In *International Conference on Learning Representations (ICLR)*, 2018. [22](#)
- [90] Adam Polyak and Lior Wolf. Channel-level acceleration of deep face representations. *IEEE Access*, 3:2163–2175, 2015. [22](#)
- [91] Xavier Suau, Luca Zappella, Vinay Palakkode, and Nicholas Apostoloff. Principal filter analysis for guided network compression. *arXiv preprint arXiv:1807.10585*, 2018. [22](#), [87](#), [88](#), [91](#)
- [92] Yuchao Li, Shaohui Lin, Baochang Zhang, Jianzhuang Liu, David Doermann, Yongjian Wu, Feiyue Huang, and Rongrong Ji. Exploiting kernel sparsity and entropy for interpretable cnn compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [22](#)

- [93] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *International Conference on Computer Vision (ICCV)*, 2017. [22](#), [87](#), [88](#), [92](#), [93](#)
- [94] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (ANIPS)*, 1990. [22](#)
- [95] Babak Hassibi and David G Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (ANIPS)*, 1993. [22](#)
- [96] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013. [22](#)
- [97] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems (ANIPS)*, 2015. [22](#)
- [98] Zelda Mariet and Suvrit Sra. Diversity networks: Neural network compression using determinantal point processes. *International Conference on Learning Representations (ICLR)*, 2016. [22](#)
- [99] Huizi Mao Song Han and William J. Dally. D. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. 2016. [22](#)
- [100] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast dpp sampling for nyström with application to kernel methods. *International Conference on International Conference on Machine Learning (ICML)*, 2016. [22](#)
- [101] Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Efficient sampling for k-determinantal point processes. *Journal of Machine Learning Research*, 2016. [23](#), [78](#), [99](#)
- [102] Yao Li, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Image co-localization by mimicking a good detector’s confidence score distribution. In *Proceedings of the European Conference on Computer Vision*, pages 19–34. Springer, 2016. [25](#), [36](#), [42](#)
- [103] Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly supervised object localization with progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3520, 2016. [25](#), [36](#), [42](#)
- [104] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004. [30](#), [32](#)

- [105] Andrew V Goldberg and Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988. [30](#)
- [106] Chloé-Agathe Azencott, Dominik Grimm, Mahito Sugiyama, Yoshinobu Kawahara, and Karsten M Borgwardt. Efficient network-guided multi-locus association mapping with graph cuts. *Bioinformatics*, 29(13):i171–i179, 2013. [30](#)
- [107] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. [32](#)
- [108] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013. [34](#)
- [109] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Minimum barrier salient object detection at 80 fps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1404–1412, 2015. [34](#)
- [110] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. The pascal visual object classes challenge 2007 (voc 2007) results (2007), 2008. [35](#), [40](#)
- [111] Parthipan Siva, Chris Russell, and Tao Xiang. In defence of negative mining for annotating weakly labelled data. In *Proceedings of the European Conference on Computer Vision*, pages 594–608. Springer, 2012. [36](#)
- [112] Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan. Weakly supervised object localization with latent category learning. In *Proceedings of the European Conference on Computer Vision*, pages 431–445. Springer, 2014. [42](#), [43](#)
- [113] Megha Pandey and Svetlana Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1307–1314. IEEE, 2011. [41](#), [42](#)
- [114] Zhiyuan Shi, Parthipan Siva, and Tao Xiang. Transfer learning by ranking for weakly supervised object annotation. *Proceedings of the British Machine Vision Conference*, pages 78.1–78.11, 2012. [36](#), [42](#)
- [115] Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 542–549. IEEE, 2012. [36](#), [38](#)
- [116] Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *Computer Vision and Pattern Recognition*,

- 2006 *IEEE Computer Society Conference on*, volume 2, pages 1605–1614. IEEE, 2006.
- [117] Zhenxing Niu, Gang Hua, Le Wang, and Xinbo Gao. Knowledge-based topic model for unsupervised object discovery and localization. *IEEE Transactions on Image Processing*, 27(1):50–63, 2018. [36](#), [41](#)
- [118] Gunhee Kim, Eric P Xing, Li Fei-Fei, and Takeo Kanade. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 169–176. IEEE, 2011. [38](#)
- [119] Parthipan Siva and Tao Xiang. Weakly supervised action detection. In *Proceedings of British Machine Vision Conference*, volume 2, page 6, 2011. [41](#)
- [120] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. *Proceedings of the European Conference on Computer Vision*, pages 452–466, 2010. [41](#)
- [121] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016. [42](#)
- [122] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *Proceedings of the European Conference on Computer Vision*, pages 350–365. Springer, 2016. [42](#), [44](#)
- [123] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. *arXiv preprint arXiv:1704.05188*, 2017. [42](#)
- [124] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. [46](#)
- [125] Herve Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. *Proceedings of the European Conference on Computer Vision*, pages 304–317, 2008. [46](#)
- [126] Florent Perronnin and Diane Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3743–3752, 2015. [47](#)
- [127] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *Proceedings of the European Conference on Computer Vision*, pages 3–20. Springer, 2016. [47](#), [48](#)

- [128] Rafael Rezendé, Joaquin Zepeda, Jean Ponce, Francis Bach, and Patrick Perez. Kernel square-loss exemplar machines for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [47](#), [48](#)
- [129] Artem Babenko and Victor Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1269–1277, 2015. [48](#)
- [130] Jianan Li, Yunchao Wei, Xiaodan Liang, Jian Dong, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Attentive contexts for object detection. *IEEE Transactions on Multimedia*, 19(5):944–954, 2017. [51](#)
- [131] Chih-Chung Hsu and Chia-Wen Lin. Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia*, 20(2):421–429, 2018.
- [132] Li Yu, Tammam Tillo, Jimin Xiao, and Marco Grangetto. Convolutional neural network for intermediate view enhancement in multiview streaming. *IEEE Transactions on Multimedia*, 20(1):15–28, 2018. [51](#)
- [133] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3): 211–252, 2015. doi: 10.1007/s11263-015-0816-y. [51](#)
- [134] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013. [51](#)
- [135] Yazhou Yao, Jian Zhang, Fumin Shen, Xiansheng Hua, Jingsong Xu, and Zhenmin Tang. Exploiting web images for dataset construction: A domain robust approach. *IEEE Transactions on Multimedia*, 19(8):1771–1784, 2017. [51](#)
- [136] Chunyan Xu, Jian Yang, Junbin Gao, Hanjiang Lai, and Shuicheng Yan. Srnn: Self-regularized neural network. *Neurocomputing*, 273:260–270, 2018. [58](#)
- [137] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pages 1062–1070, 2015. [61](#), [62](#), [63](#), [70](#)
- [138] Abrar H Abdalnabi, Gang Wang, Jiwen Lu, and Kui Jia. Multi-task cnn model for attribute prediction. *IEEE Transactions on Multimedia*, 17(11): 1949–1959, 2015. [61](#)

- [139] Alex Krizhevsky and G Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 2010. [62](#)
- [140] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5. Granada, Spain, 2011. [62](#)
- [141] Ishan Jindal, Matthew Nokedby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 967–972, 2016. [63](#), [68](#), [69](#), [71](#), [72](#)
- [142] Xin Liu, Shaoxin Li, Meina Kan, Shiguang Shan, and Xilin Chen. Self-error-correcting convolutional neural network for learning with noisy labels. In *Proceedings of the IEEE International Conference on Automatic Face & Gesture Recognition (FG)*, pages 111–117, 2017. [63](#), [68](#)
- [143] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. [63](#), [68](#), [69](#)
- [144] Amnon Drory, Shai Avidan, and Raja Giryes. On the resistance of neural nets to label noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. [63](#), [68](#), [69](#)
- [145] Chenghua Li, Chunjie Zhang, Kun Ding, Gang Li, Jian Cheng, and Hanqing Lu. Bundlenet: Learning with noisy label via sample correlations. *IEEE Access*, 6:2367–2377, 2018. [63](#), [68](#)
- [146] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. [63](#), [68](#), [69](#), [71](#), [72](#)
- [147] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. [63](#)
- [148] Weixiang Hong, Junsong Yuan, and Sreyasee Das Bhattacharjee. Fried binary embedding for high-dimensional visual features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2749–2757, 2017. [74](#)
- [149] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *International Conference on Computer Vision (ICCV)*, 2017. [74](#), [77](#)

- [150] Jingyi Hou, Xinxiao Wu, Yuchao Sun, and Yunde Jia. Content-attention representation by factorized action-scene network for action recognition. *IEEE Transactions on Multimedia*, 20(6):1537–1547, 2018. [74](#)
- [151] Pichao Wang, Wanqing Li, Zhimin Gao, Chang Tang, and Philip O Ogunbona. Depth pooling based large-scale 3-d action recognition with convolutional neural networks. *IEEE Transactions on Multimedia*, 20(5):1051–1061, 2018. [74](#)
- [152] Jinzhuo Wang, Wenmin Wang, and Wen Gao. Multiscale deep alternative neural network for large-scale video classification. *IEEE Transactions on Multimedia*, 20(10):2578–2592, 2018. [74](#)
- [153] Yu-Gang Jiang, Zuxuan Wu, Jinhui Tang, Zechao Li, Xiangyang Xue, and Shih-Fu Chang. Modeling multimodal clues in a hybrid deep learning framework for video classification. *IEEE Transactions on Multimedia*, 20(11):3137–3147, 2018.
- [154] Tan Yu and Junsong Yuan. Compressive quantization for fast object instance search in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. [74](#)
- [155] Hao Zhou, Jose M Alvarez, and Fatih Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision (ECCV)*, 2016. [77](#), [78](#)
- [156] Sajid Anwar and Wonyong Sung. Compact deep convolutional neural networks with coarse pruning. *arXiv preprint arXiv:1610.09639*, 2016. [78](#), [83](#)
- [157] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. Accelerating very deep convolutional networks for classification and detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38(10):1943–1955, 2016. [80](#)
- [158] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision (ICCV)*, 2015. [82](#), [83](#)
- [159] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *International Conference on Computer Vision (ICCV)*, 2017. [82](#), [83](#)
- [160] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [82](#)
- [161] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [82](#)

- [162] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [82](#), [84](#)
- [163] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009. [82](#)
- [164] Cagdas Bak, Aysun Kocak, Erkut Erdem, and Aykut Erdem. Spatio-temporal saliency networks for dynamic saliency prediction. *IEEE Transactions on Multimedia*, 20(7):1688–1698, 2018. [83](#)
- [165] Naoya Takahashi, Michael Gygli, and Luc Van Gool. Aenet: Learning deep audio features for video analysis. *IEEE Transactions on Multimedia*, 20(3): 513–524, 2018.
- [166] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055, 2017. [83](#)
- [167] Jian-Hao Luo, Hao Zhang, Hong-Yu Zhou, Chen-Wei Xie, Jianxin Wu, and Weiyao Lin. Thinet: pruning cnn filters for a thinner net. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2018. [88](#)
- [168] Qiangui Huang, Kevin Zhou, Suyu You, and Ulrich Neumann. Learning to prune filters in convolutional neural networks. *International Workshop on Applications of Computer Vision (WACV)*, 2018. [91](#)
- [169] Yiming Hu, Siyang Sun, Jianquan Li, Xingang Wang, and Qingyi Gu. A novel channel pruning method for deep neural network compression. *arXiv preprint arXiv:1805.11394*, 2018. [91](#)
- [170] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient transfer learning. *International Conference on Learning Representations (ICLR)*, 2016. [92](#)
- [171] Shaohui Lin, Rongrong Ji, Yuchao Li, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Accelerating convolutional networks via global & dynamic filter pruning. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2018. [92](#), [93](#)
- [172] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. [98](#)
- [173] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. [98](#)