Web text-aided image classification

Wang, Dongzhe

2019

Wang, D. (2019). Web text-aided image classification. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/105855>

<https://doi.org/10.32657/10220/47855>

Web Text-Aided Image Classification

Dongzhe Wang

School of Electrical & Electronic Engineering

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

2019

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.

18-03-2019

.....

Date

Dongzhe Wang

.....

Dongzhe Wang

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.

18-03-2019

.....

Date

Kezhi Mao

.....

Prof. Kezhi Mao

Authorship Attribution Statement

This thesis **does not** contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

18-03-2019

.....

Date

Dongzhe Wang

.....

Dongzhe Wang

Acknowledgements

First of all, I would like to express my sincere gratitude to my PhD supervisor, Prof. Kezhi Mao, who provided me great guidance and academic freedom throughout my PhD study in the past four years. His expertise and insights in the field of machine-learning have been a great help for me to explore the nature behind those challenging research problems. To accomplish solid research work, Prof. Mao has spent a large amount of time discussing research methodology and directions with me. All the methods and ideas that presented in this thesis are impossible without his invaluable guidance, constant support and patience. He has been the biggest source of encouragement for this work.

I would also give my thanks to my great master advisor and co-advisor Prof. David Miller and Prof. George Kesidis, both from the department of Electrical Engineering at the Pennsylvania State University, USA. They have opened an window for me to have a taste of the beauty of pattern recognition and machine-learning research. Prof. Miller and Prof. Kesidis also encourage me to pursue the PhD degree following machine-learning path.

Meanwhile, I wish to thank Dr. Rui Zhao, Dr. Zhenhua Chen, Dr. Le Zhang, Dr. JinMing Xu, and Dr. Xuefeng Yang. These senior PhD colleagues gave me a lot of constructive suggestions in research and life in the past few years. Without these awesome colleagues, this thesis would not have been possible. Special thanks to all my amazing friends in Internet of Things Laboratory and Singapore generally for their encouragement, care, and friendship: Dr. Zuo Bai, Dr. Zhi Zhou, Dr. Yong Zhang, Dr. Jie Ding, Dr. Qingchang Zhu, Qinxu Ding, Pengfei Li, Xuhao, Li, Dr. Leibin Ni, Shihui Liu, Zhaokai Zhu, Jinhan Li, Yupeng Li and so many friends. We spend very joyful time in and out the lab and my dear friends made my graduate life in Singapore colorful.

Last but not least, I would also like to thank my family for their unconditional love. My parents have supported and understood me throughout the several years

it has taken to complete my long study journey. My lovely wife Mengru has been hugely supportive on all my decisions since we have met. They are certainly the best in my life and I love them so much.

NLP		Natural Language Processing
BoW		Bag-of-Words
BoF		Bag-of-Features
PHOW		Pyramid Histogram Of visual Words
SIFT		Scale-Invariant Feature Transform
FV		Fisher vector
SPM		Spatial Pyramid Matching
LDA		Linear Discriminative Analysis
SC		Sparse Coding
VQ		Vector Quantization
PLSA		Probability Latent Semantic Analysis
CNN		Convolutional Neural Network
LSTMs		Long-short Term Memory Networks
LLC		Locality-constrained Linear Coding
CCLR		Correlation Constrained Low Rank
ILSVRC		Large Scale Visual Recognition Challenge
VOC		Visual Object Classes
TF-IDF		Term Frequency-Inverse Document Frequency
DNN		Deep Neural Network
SMNN		Semantic Matching Neural Network
mm-CNN		Multimodal Convolutional Neural Network
PCA		Principal Component Analysis
s-CNN		Semantic Convolutional Neural Network
s-CNNC		Semantic Convolutional Neural Network Clustered

AAN		Adaptive Attention Network
LSA		Latent Semantic Analysis
SGD		Stochastic Gradient Descent
MLP		Multilayer Perceptron
BP		Backpropagation
VSM		Vector Space Model
CBoW		Continuous Bag-of-Words
RNN		Recurrent Neural Networks
GRU		Gated Recurrent Units
SVM		Support Vector Machine
ReLU		Rectified Linear Unit
CCA		Canonical Correlation Analysis
VGG		Visual Geometry Group
AP		Average Precision
mAP		mean Average Precision
t-SNE		t-Distributed Stochastic Neighbor Embedding
LCS		Least Common Subsume
RBF		Radial Basis Function
mAcc		mean Classification Accuracy
DDSFL		Deep Discriminative and Shareable Feature Learning
ZSL		Zero-Shot Learning
LM		Language Model
ELMo		Embeddings from Language Models (ELMo)
ULMFiT		Universal Language Model Fine-tuning

Abstract

Image classification is often solved as a machine-learning problem, where a classifier is first learned from training data, and class labels are then assigned to unlabeled testing data based on the outputs of the classifier. To train an image classifier with good generalization capability, conventional methods often require a large number of human labeled training images. However, a large number of well-labeled training images may not always be available. With the exponential growth of web data, exploiting multimodal online sources via standard search engine has become a trend in visual recognition as it can effectively alleviate the shortage of training data. However, the web data such as text data is often not cooperative due to its unstructured and noisy nature. Therefore, how to represent and utilize the web text data to aid image classification is chosen as the focus of this thesis.

Since target image data and web text data are usually from different domains whose representations are in the different feature space, we firstly investigate the two modalities of data separately and then combine the bimodal information in decision level. In particular, low-level text modeling approaches including class tag occurrence and bag-of-words vectorization and image modeling approaches such as dense SIFT are employed to learn separate classifiers, whose decision scores are aggregated adaptively. On the other hand, we believe that the correlation information between image modality and web text modality is also very important. In order to explore the cross-modal correlation, we also investigate feature-level multimodal fusion models in this PhD thesis. Learning dense and real-valued text representation in a similar manner of learning image representation is the keystone of feature-level multimodal fusion. In this thesis, we propose the novel task-specific semantic matching network and task-generic semantic convolutional network models to learn semantic text features. These proposed text feature learning methods are motivated by the transferable mid-level image representation learned by the convolutional neural network (CNN). Besides traditional supervised learning setting, we find that the web text-aided strategy also makes difference in weakly

supervised setting when only little labeled data is available. Specially, we investigate web text-aided one-shot learning that is able to identify unlabeled data from novel classes based on single observation using an adaptive attention mechanism.

This thesis is organized as follows. Chapter 1 introduces the motivation behind the web resources-aided image classification. Chapter 2 reviews the related works in this field, including image representation learning, text representation learning and multimodal fusion learning.

Chapter 3 investigates the decision-level data fusion for web-aided image classification. An adaptive combiner for two separate bimodal classifiers is developed in decision level. This adaptive fusion algorithm is inspired by the multisensory integration mechanism of human. And the adaptability is achieved by reliability-dependent weighting of different sensory modalities.

In Chapter 4, a novel text modeling namely the semantic matching neural network (SMNN) is proposed, which is quantified by cosine similarity measures between embedded text input and task-specific semantic filters. It is capable of learning semantic features from the associated text of web images. The SMNN text features have improved reliability and applicability, compared to the text features obtained from other methods. Then, the SMNN text features and convolutional neural network visual features are jointly learned in a shared representation, which aims to capture the correlations between the two modalities in the feature level.

Improving upon task-specific filters for SMNN, Chapter 5 presents a novel semantic CNN (s-CNN) model for high-level text representation learning to encode semantic correlation based on task-generic semantic filters. However, the s-CNN model inevitably brings about surplus semantic filters to achieve better applicability and generalization in universal tasks. Moreover, the surplus filters may lead to semantic overlaps and feature redundancy issue. To address this issue, the s-CNN Clustered (s-CNNC) models that uses filter clusters instead of individual filters is presented. Interacting with the image CNN models, the s-CNNC models can further boost image classification under a multimodal framework, which can be trained end-to-end.

Chapter 6 develops an adaptive encoder-decoder attention network that uses web text to aid one-shot image classification. Without any ground truth semantic clues, e.g., class tag information, our model is able to extract useful information

from web sourced data instead. To address the noise nature of web text, the adaptive mechanism is introduced to determine when to attend to text-inferred visual features and when to rely on original visual features.

The summarization and future prospect of my PhD work is finally discussed in Chapter [7](#).

Contents

Acknowledgements	i
Acronym List	iii
Abstract	v
List of Figures	xiii
List of Tables	xvii
1 Introduction	1
1.1 Overview	1
1.2 Contributions and Thesis Outline	5
2 Literature Review	11
2.1 Image Representation Learning	11
2.1.1 Bag-of-Features Image Representation	12
2.1.2 Deep Feature Representation	15
2.2 Text Representation Learning	20
2.2.1 Bag-of-Words Representation	20
2.2.2 Word Embeddings	23
2.2.3 CNN Representation	24
2.3 Multimodal Data Fusion	27
2.3.1 Multi-View-Based Data Integration	27
2.3.2 Neural Network-based Data Integration	29
2.4 Summary of Related Work	29
3 Web Text-Aided Image Classification Based on Adaptive Decision-Level Fusion	31
3.1 Introduction	32
3.1.1 Motivations	32
3.2 Web Text Representation Learning	33
3.2.1 Online Web Text Retrieval and Data Preprocessing	33
3.2.2 Class Label-Based Text Representation	36

3.2.3	BoW-Based Text Representation	36
3.3	Adaptive Fusion based on Multimodal Image Classification	39
3.3.1	Web Text-Based Classification	39
3.3.2	Training Image-Based Classification	40
3.3.3	Adaptive Multimodal Fusion	41
3.4	Experiments and Results	43
3.4.1	Compared Approaches	45
3.4.2	Experimental Setups and Evaluation Metrics	46
3.4.3	Results of UIUC-Sports dataset	47
3.4.4	Results of Scene-15 dataset	48
3.4.5	Results of Caltech-256 dataset	50
3.4.6	Results on VOC-2012 dataset	51
3.5	Conclusions	53
4	Web Text-Aided Image Classification by Incorporating Task-Specific Semantic Matching Neural Network	55
4.1	Introduction	56
4.1.1	Motivations and Overview	56
4.1.2	Text Filtering Methods	57
4.2	Semantic Neural Network	58
4.2.1	Network Architecture	58
4.2.2	Generation of Task-Specific Semantic Filters	60
4.2.3	Difference Between SMNN and CNN	63
4.2.4	Visual-Text Feature-Level Fusion	63
4.3	Experiments	64
4.3.1	Experiment Setup	64
4.3.2	Results: Single-Label Classification	66
4.3.3	Results: Multi-Label Classification	69
4.3.4	Results: Semantic Filters	71
4.4	Conclusions	71
5	Task-Generic Representation Learning Method Based on Semantic Convolutional Neural Network for Web Text-Aided Image Classification	73
5.1	Introduction	74
5.1.1	Motivations	74
5.1.2	Semantic Feature Learning	75
5.2	Semantic Convolutional Neural Network	75
5.2.1	Generation of Task-Generic Semantic Filters	77
5.2.2	s-CNN with Double Pooling	81
5.3	Multimodal CNN	85
5.3.1	Multimodal CNN for Joint Representation Learning	85
5.3.2	Multimodal CNN with Noise Reduction	85
5.4	Experiments	86

5.4.1	Datasets	87
5.4.2	Experimental Setup and Evaluation Metrics	88
5.4.3	Results on Semantic Convolutional Neural Network	90
5.4.4	Results on Scene Recognition	91
5.4.5	Results on Object Recognition	92
5.5	Conclusions	96
6	Adaptive Attention Network for Web Text-Aided One-Shot Image Classification	97
6.1	Introduction	98
6.1.1	Motivations	98
6.1.2	One-Shot Learning	100
6.1.3	Attention Mechanism	100
6.2	Proposed Method	101
6.2.1	Problem Definition	101
6.2.2	Web Text-Aided Attention Model	102
6.2.3	Adaptive Attention Model	105
6.3	Experiments	107
6.3.1	Datasets	107
6.3.2	Experimental Setup	108
6.3.3	Results on VOC-2007 Subsets	110
6.3.4	Results on MIT-Indoor67 Subset	111
6.3.5	Attention Maps Visualization	112
6.4	Conclusion	113
7	Conclusion and Future Work	115
7.1	Conclusion	115
7.2	Future Directions	117
	Author's Publications	119
	Bibliography	121

List of Figures

1.1	Retrieval examples by image queries with Google reverse image search on Scene-15: “Coast” and “Highway”. Here, we only show the top 9 returned results.	3
1.2	A paradigm of decision-level fusion methods	5
1.3	A paradigm of feature-level fusion methods	6
2.1	The overview of the baseline SPM model based on BoF method. . .	13
2.2	An example of PHOW descriptors with two levels of pyramid, where $L = 2$	14
2.3	The overall structures of AlexNet with 5 convolutional (conv) layers and 3 fully connected (FC) layers.	16
2.4	The overall structures of Network In Network with the stacking of three multilayer perceptron convolutional (mlp_conv) layers and one global average pooling (GAP) layer.	16
2.5	An example diagram of GoogLeNet with “Inception” modules. The Inception modules combine convolutions of different sizes. Note that the multiple softmax at different inception modules are not shown due to the space.	17
2.6	An example residual block in the ResNet.	18
2.7	The framework for CNN transferring and Fine-tuning using pre-trained VGG as example. FC_a is an adaptation layer and FC_b is the output layer of the new task.	19
2.8	A toy example on the one-hot encoding method.	20
2.9	The framework for Neural Language Model (LM).	23
2.10	An architecture of one convolutional layer CNN with word embedding input for text representation.	25
2.11	The architecture of RNN (with 3 time steps) encoder and GRU encoder.	27
2.12	Subspace learning framework for image-text data fusion.	28
3.1	Two similar but different scenes, “bocce” and “croquet”.	33
3.2	An overview of the proposed framework with adaptive decision-level fusion algorithm.	34

3.3	An example of the procedure of online web text search (upper half) and data preprocessing (lower half) with the class label “climbing” when $n = 4$. The class label information in the image captions and descriptions is circled in red.	35
3.4	An overview of the VSM-based noise filtering.	37
3.5	Example images for the UIUC-Sports dataset.	47
3.6	Example images for the Scene-15 dataset. The starred 8 categories are the originally obtained from [1] (a.k.a., LabelMe8 dataset). . . .	49
3.7	Example images for the VOC-2012 dataset.	52
4.1	An overview of the proposed web aided neural network-based fusion framework. Here the online web retrieval process of a test image is shown, and we do likewise for training images to extract web text data.	57
4.2	Illustration of a Semantic Neural Network architecture for text classification: semantic-filter layer and pooling layer. The boxes (in different colors) of the semantic-filter layer represent different separate feature maps (sets of filter responses). The output text feature representation is finally fed to the classifier.	59
4.3	T-SNE visualization of web text data of UIUC-Sports. The web text data is represented by three different text features: (a) BoW (TF-IDF) feature, (b) CNN feature [2] (wrapped with word embeddings), and (c) our proposed SMNN feature. The class labels of data samples are denoted in the color bar on the right.	61
4.4	Flowchart of the semantic filter bank formulation. Semantic filters are word embeddings of prior knowledge-based representative words that best describes the image classes.	62
5.1	System overview. Our proposed method consists of two main modules: transferring and fine-tuning on the pre-trained CNN model and learning high-level text representation via s-CNN with filter clustering (framed in the shades). The aggregation of the two modules is performed by training a unified neural network. This figure is best viewed in color.	76
5.2	Contrastive s-CNN _{max} experiments to find m within $\{Q-100, Q, Q+100\}$. (a): object recognition; (b): scene recognition.	89
6.1	Overview of the semantic-level feature extraction and encoding. The four examples of web visually similar images are obtained using online-based image search [3], and the text descriptions of them are aggregated and represented by pre-trained word vectors.	102
6.2	An illustration of our proposed spatial attention network.	105
6.3	An illustration of our proposed adaptive attention network.	107
6.4	Visualization of attention maps of MA method [4] on the VOC subset 2 novel classes, where the intensity indicates the activeness of different regions.	113

6.5	Visualization of attention maps of our proposed adaptive attention model on the novel classes, where the β values (shown as “b”) are displayed at the upper left corners of the attention maps.	114
-----	---	-----

List of Tables

2.1	Common nonlinear activation functions in CNN. “tanh”: hyperbolic tangent; “ReLU”: rectified linear units.	26
3.1	Statistics of datasets (note that we show statistics of VOC12 validation set only due to the unavailability of test labels). N : Number of images. $ V $: Vocabulary size. $ V_{abs} $: Size of vocabulary that absent in the pre-trained word vectors.	45
3.2	Mean accuracy (%) of the state-of-the-art feature extractors (average over 30 trials) on UIUC-Sports.	47
3.3	Per-class accuracy (%) comparison of the model using visual descriptors only, text descriptors only, and the proposed decision-level fusion models on UIUC-Sports.(average over 30 trials).	48
3.4	Classification accuracy (%) of the state-of-the-art feature extractors (average over 30 trials) on Scene-15 Dataset.	50
3.5	Classification accuracy (%) on the Caltech-256 dataset. The number of “15” and “60” in the table denotes the number of data for training.	50
3.6	Per-class AP (%) of the decision-level fusion methods on VOC-2012 dataset. Note that the fusion methods shown in the last two rows of the table are wrapped with VGG16 visual features.	52
4.1	Classification mAcc results (%) with different text features on the auxiliary web text data of UIUC-Sports, Scene-15 and Caltech-256 datasets. The shown results on Caltech-256 are obtained using the 60 images/category setting.	66
4.2	Comparison of Top-1 mAcc (%) between our methods and other state-of-the-art methods on UIUC-Sports, Scene-15, and Caltech-256 datasets. The starred methods show the results reported in Chapter 3.	67
4.3	Per-class AP (%) comparison between our proposed SMNN methods and other text features methods on the auxiliary web text data of VOC-2012 dataset.	69
4.4	Per-class AP (%) comparison between our proposed web aided neural network-based fusion methods and the baseline results on VOC-2012 dataset. The starred method demonstrates the results reported in Chapter 3.	70

4.5	Examples of the some semantic filters for three datasets. We list the class labels of three dataset in the second column, which are the semantic queries to obtain the web reference documents. The filter names shown in the third column are selected from the preliminary vocabulary without their synonyms obtained from WordNet.	72
5.1	Semantic filters for object recognition and scene recognition. Four exemplars per category are shown in WordNet format.	79
5.2	Comparison of top-1 accuracy (%) to the comparing approaches on the text representation of 4 different datasets. The first three datasets are scene-centric benchmarks, while the last one is an object-centric dataset. The starred methods are the methods shown in Chapter 4.	90
5.3	Comparison of mAP (%) to the comparing approaches on the text representation of VOC-2007 and VOC-2012 object-centric datasets.	91
5.4	Top-1 classification accuracy (%) on the Scene-15, MIT Indoor-67 and UIUC-Sports datasets for the state-of-the-art methods and our methods. The starred result is the best performance shown in Chapter 4.	92
5.5	Top-1 Classification accuracy (%) on the Caltech-256 dataset. Note that the method listed with ^ denotes that only a “subset” of web text data is included, which is selected by the mm-CNN with noise reduction.	93
5.6	Per-class average precision (%) comparison for the listed methods on VOC-2007 test dataset. Note that the method listed with ^ denotes that only a “subset” of web text data is used.	94
5.7	Per-class average precision (%) comparison for the listed methods on VOC-2012 test dataset. The starred method denotes the methods shown in Chapter 4.	95
6.1	Overall mAP (%) comparison of the proposed adaptive attention network to the other baseline methods on VOC-2007: (A) subset 1 , (B) subset 2 The number in the parenthesis indicates the number of attention maps involved. The last two columns in the tables represent the close-world and open-world experimental settings, respectively.	111
6.2	Overall mAP (%) comparison of the proposed adaptive attention network to the other baseline methods on MIT-Indoor67 subset. . .	112

Chapter 1

Introduction

1.1 Overview

As one key task of supervised learning problems, image classification has received considerable attention in computer vision recently, where a classifier is firstly learned from training data, and then applied to test data to predict its label based on the outputs of classifier. The classification performance largely depends on the availability of sufficient well-labeled data for training. More recently, deep neural networks such as convolutional neural networks (CNN) have achieved remarkable improvements over the low-level image modeling methods by a large margin in many visual recognition tasks. The success of CNN for modeling images is credited to the emergence of the large-scale image repositories such as ImageNet [5]. No matter deep or not, the performance of image classification is always positively correlated with the size of training data. However, sufficient amount of training labeled data demands labor-expensive and time-consuming human annotations.

Meanwhile, a variety of open source data has become available due to the advent and explosive growth of world-wide web. According to the statistics, more than 280 thousand new photos with users' annotations are uploaded to Flickr alone every minute. Unlike the large-scale human labeled data that is scarce and difficult to obtain, we can always get abundant and cheap web data easily from open sources. More importantly, the open source data can truly help address the training data shortage problem for specific tasks if it is properly used. For example, several web

homogeneous¹ data-aided image classification approaches have been proposed, including self-taught learning [6], domain adaptation [7], semi-supervised learning [8] and etc. These existing methods utilize some unlabeled images that have the same feature representation from the same domain as the source data to address the shortage of labeled data. Another common characteristic of these approaches is the class keyword-based web search, which downloads task-specific images to construct the web image database. By taking the web images as normal training images, the size of labeled image data is expanded, and this in turn helps build the classifiers with better generalization. The use of external image data indeed relieves the shortage of training data, but these works only use the visual features of the web images, without using high-level semantic textual feature carried by the tags or captions of the web images.

Different from homogeneous web data-aided approaches, heterogeneous web data-aided frameworks [9–11] have been developed to explore different modality data and facilitate image classification, such as image tags or descriptions in the form of short text. Compared to homogeneous frameworks, heterogeneous frameworks not only use the extra images that have the same feature representation for training, but also investigate different feature representations for the web text information. Similar to the homogeneous data-aided frameworks, the task-specific web image databases are usually pre-constructed for the most existing heterogeneous data-aided frameworks. Based on the similarity ranking of web images with respect to the local images, some studies such as [9] and [12] were able to find the visually similar images of the local images from the pre-constructed web image databases. Their associated text can be used as the supplementary text data of local images, from which the text features are learned to aid image classification. Although these heterogeneous aided frameworks have been proven to be effective, their text retrieval process has some shortcomings. The authors in [9] used large size of the web image database (over 1 million images) to pursue a good coverage of the test images, which requires massive local storage space and consumed enormous computing power. In addition, the class keyword-based web search strategy completely relies on the class tag information to build up the alignment between specific images and text, which may not exist in practice.

¹The training data and web resources are in the same modality

Image query	Google retrieved 9 most visually similar images and annotations
<p>Z-Enterprises Group, Luxury Yachting Greece Experience, Yachts, Support, Yacht Charter</p>	<p>Super yacht 'Ionian Princess' Special Offer in Greece</p>	<p>Motor Yacht Barents Sea HELLAS yacht Luxury Private Yacht Charter Greece</p>
	<p>Grimaldi Launches New Maritime Service Between Italy and Greece</p>	<p>Super yacht KJO available for charter in Croatia, Greece or Turkey</p>	<p>O'MEGA Mega Yacht Charter Greece</p>
	<p>Luxury Mega yacht charter Greece</p>	<p>GOLD COAST CHARTER BOATS Luxury Boat Charters</p>	<p>East Coast Yacht Sales - Yarmouth Maine</p>
<p>Planning your road trip on a budget</p>	<p>American highway / road pictures Picture of American highway / road - Free Pictures</p>	<p>Iowa Makes a Bold Admission: We Need Fewer Roads</p>
	<p>File:RybnoeDistrict 06-13 Road to Konstantinovo.jpg</p>	<p>Road, Objective, Turning Point, Unknown</p>	<p>Endless Road by friedapi on DeviantArt by friedapi</p>
	<p>Focus On: Building Infrastructure in Africa South Africa Road</p>	<p>A straight two lane highway crosses the desolate Nevada</p>	<p>Highway Through The Countryside Of Eastern Washington</p>

FIGURE 1.1: Retrieval examples by image queries with Google reverse image search on Scene-15: “Coast” and “Highway”. Here, we only show the top 9 returned results.

In contrast, we propose an online search-based text retrieval method for web text-aided image classification. Intuitively, we find that it is more straightforward and convenient to obtain the web text of the query images by utilizing the powerful search engine of the Google reverse image search. The merits of our online search-based method are summarized as follows. Firstly, we regard the entire world-wide web as a gigantic database, and thus the size of such database should be large enough to cover sufficient visually similar web images. Moreover, the returned text data from the web are not limited to specific image class names or tags. Secondly, we directly crawl and collect the web text of the top n most visually similar web images of every query image without downloading the images, where n is set by users. This will significantly save local storage space and computations with the help of Google’s millions of servers and online storage systems. The Google-based online image-to-text retrieval examples are exhibited in Figure 1.1.

In spite of different web-based search methods for text retrieval, an important question must be asked once the web text is applied to image classification: how to aggregate the heterogeneous data, i.e., text and image, in order to improve image classification? In this PhD thesis, we attempt to find the solutions to this question from the perspective of multimodal data fusion strategies and advanced text representation learning methods. Since more semantic information has been included in our work, the proposed methods should be able to:

- preserve the independence of different modalities and incorporate their respective discriminative powers in high level;
- learn dense and real-valued high-level text representations, in a similar way of learning the image representation from the pre-trained image CNN baselines;
- explore the semantic correlation underlying different modalities in a joint representation;
- generalize and extend to weakly supervised image classification problems, such as one-shot image classification.

To address the objectives above, we present our achievement and organize the outline of this thesis in the following section.

FIGURE 1.2: A paradigm of decision-level fusion methods

1.2 Contributions and Thesis Outline

The main contribution of my thesis is to use abundant and publicly accessible text data to relieve the limitation of scarce and expensive training data for image classification. To begin with, the prerequisite of the methods presented in this thesis is the online-search text retrieval approach, which helps to get rid of the dependency on the task-specific class labels. By exploring the web text data using different representation learning methods, we propose two types data fusion frameworks in this thesis from the perspective of decision level and feature level, respectively. These fusion frameworks integrate the cross-modal information underlying the image and web text. Moreover, we find that the web text data can also benefits a newly emerging classification problem: one-shot learning, which requires rapid adaption to the unlabeled data using the visual clues learned from a single labeled observation. We manage to bridge the semantic gap between the weakly supervised

FIGURE 1.3: A paradigm of feature-level fusion methods

information and the unlabeled data using an adaptive attention mechanism based on the web text data.

In Chapter 2, the related work in the web text aided image classification are reviewed including image representation, text representation, and multimodal data integration. In Chapter 3, Chapter 4 and Chapter 5, three multimodal fusion frameworks and two novel text representation learning methods are exhibited. Chapter 6 extends our web text-aided scheme to a more challenging one-shot learning problems. Lastly, a brief conclusion and profound discussion of this thesis are demonstrated with the potential future research directions in Chapter 7. More details of the proposed methodologies in this thesis are discussed as follows:

- *Chapter 3: Web Text-Aided Image Classification Based on Adaptive Decision-Level Fusion:*

The method introduced in this chapter is our first attempt to use web resources to aid visual recognition tasks. In contrast to image data, web resources (e.g., text) are obtained from different domain and represented in different feature space. To address this modality heterogeneity, we regard multiple modalities separately as different views of image data so that the multimodal structure is not affected by the vulnerable interaction of heterogeneous feature space. In particular, the bimodality data is characterized by some classical and off-the-shelf text and image modeling approaches for their classifiers training, such as the bag-of-words (BoW)-based model and the dense SIFT model for text representation learning and image representation learning, respectively. The strength of decision-level fusion framework reflects in the fact that it can further boost the original classification performance after the classifier training is done. Next, considering the reliability diversity of web resources, the two modalities of a testing data are concatenated by reliability-dependent weighting modulation within an adaptive fusion architecture. The main goal of this framework is to prove the idea of web text-aided classification is feasible, as shown in Figure 1.2. In overall, our proposed method explicitly validates the effectiveness of using web text for improved image classification. This part of work is published in paper “Adaptive Multimodal Fusion with Web Resources for Scene Classification” [13].

- *Chapter 4: Web Text-Aided Image Classification by Incorporating Semantic Matching Neural Network:*

The adaptive decision-level fusion system preserves the information of multimodal structure, whereas it dismisses the underlying correlations across multiple modalities. To address this potential limitation, we explore an unified feature-level multimodal fusion framework by learning a mid-level feature representation for image classification in this chapter, whose diagram has been shown in Figure 1.3. To this end, we develop a textual modeling method named semantic matching neural network (SMNN) that is capable of learning semantic features from the associated text of web images. Our proposed SMNN is structurally similar to CNN but with layers of *task-specific* word filters, which are the word vectors selected from web knowledge base with respect to the image class labels. Therefore, the filtering process of SMNN is essentially a semantic matching process based on word-wise correlation quantified by cosine similarity measurement between word vectors. In

addition, the weights of words filters are assigned directly from pre-trained language models without additional training. Hence, the semantic neural network training is computationally very efficient. This part of work is published in paper “Learning Semantic Text Features for Web Text Aided Image Classification” [14].

- *Chapter 5: Task-Generic Representation Learning Method Based on Semantic Convolutional Neural Network for Web Text-Aided Image Classification:*

The proposed work in this chapter is our second exploration on unified feature-level multimodal fusion framework. Unlike the SMNN proposed in Chapter 4 that involves task-specific text filters in network training, we propose the semantic convolutional neural network (s-CNN) model for high-level text representation learning using *task-generic* semantic filters. The proposed s-CNN model is able to extract task-generic features and learn transferable text representation, in a similar manner of image CNN baselines. Since our model inevitably brings about surplus semantic filters to achieve a good coverage of universal tasks, we develop the so-called s-CNN Clustered (s-CNNC) model that can effectively solve the feature redundancy problem caused by the semantic overlaps of the filters. Interacting with the image CNN models, we develop a noise reduction scheme for the multimodal CNN fusion model to alleviate the negative impact of the noisy components underlying the web text data. This part of work is published in paper “Task-Generic Semantic Convolutional Neural Network for Web Text-Aided Image Classification” [15].

- *Chapter 6: Adaptive Attention Network for Web Text-Aided One-Shot Image Classification:*

In this chapter, we extend our exploration of web text-aided framework to a more challenging specific problem: one-shot learning. Unlike strongly supervised learning prototypes, one-shot learning has the extraordinary ability to recognize unfamiliar images from a single observation with labeling information. We show that informative web resources can be employed to facilitate image classification in one-shot setting without knowing the class tags of the novel classes. We design an adaptive attention model to decide whether to attend to the visual clue informed by the web text or to the image content itself. We validate the performance of our proposed attention network on both object and scene classification problems, and our methods outperform

the baseline methods significantly. Under the one-shot settings, we show the practical importance of adaptive web text-aided attention model in classifying both the object-centric images and the more complex indoor scene-centric images. This part of work is published in paper “Adaptive Web Text-Aided Attention Network for One-Shot Image Classification” [\[16\]](#).

Chapter 2

Literature Review

In this chapter, we systematically review the state-of-the-art work that is related to our proposed methods, from the perspective of image representation learning, text modeling, and the multimodal learning methods. To begin with, we start our review from classical shallow image representations and hotly explored deep image representations in Section 2.1. In Section 2.2, we discuss the state-of-the-art text representation learning methods from low-level to high-level models. Many of these image and text representation learning approaches will be exploited in our proposed methods in Chapter 3, Chapter 4, Chapter 5 and Chapter 6. In Section 2.3, a brief review of multimodal fusion frameworks based on image modality and text modality is given. Inspired by the models and methods reviewed in this chapter, we draw some insights and develop our work in turn.

2.1 Image Representation Learning

Visual recognition has a wide variety of applications including image classification. Image classification refers to the task of assigning a class label to an image. The keystone of the image classification problem is to attain effective representations of images. Then, the learning algorithms including Support Vector Machine [17], logistic regression, random forest, linear discriminative analysis (LDA), Naive Bayes classifier, probabilistic latent semantic analysis (PLSA), and many more methods are adopted to perform classification on the learned image representations. For

instance, one of the most popular classifiers in traditional machine learning techniques is SVM because it can control the model complexity independent of the dimensionality of input space. The core idea of SVM classifiers is to find the separating hyperplane (decision boundary) that maximizes the margin (the minimum distance between the decision boundary and any of the samples). In linear separable SVM situations, the discriminant function is derived as

$$g(x) = w^T \phi(x) + b, \quad (2.1)$$

where $\phi(x)$ denotes a fixed feature space transformation and b the bias parameter. The maximum margin solution is found by solving

$$J(w, b) = \arg \max_{w, b} \left\{ \frac{1}{\|w\|^2} \min_n [\text{sign}(w^T \phi(x) + b)] \right\}, \quad (2.2)$$

The optimal hyperplane subject to the constraint $\text{sign}(w^T \phi(x) + b) \geq 1$. In order to solve this constrained optimization problem, the Lagrange multipliers $\alpha_n \geq 0$ are introduced. To classify a new data, the new discriminant function

$$g(x)^* = \sum_{n=1}^N \alpha_n \text{sign}(K(w, x)) + b \quad (2.3)$$

satisfies the Karush-Kuhn-Tucker (KKT) conditions [18]. By this way, we use the previously obtained representation of training images to train the SVM model, which is then applied to predict the classes of testing images. In the following sub-sections, we will review the development of image representations.

2.1.1 Bag-of-Features Image Representation

In the past few decades, various image descriptors have been developed based on *bag-of-features* (BoF) (a.k.a, *bag-of-visual words*) method. The BoF-based models are originated from the *bag-of-words* models in document analysis, but the BoF-based models emphasize the texon frequency for texture recognition from the perspective of image patches. In particular, an image is usually represented as dense and real-valued feature vectors in BoF-based models. In general, the BoF-based models are obtained by four steps as follows:

FIGURE 2.1: The overview of the baseline SPM model based on BoF method.

- Extract features from a training set, which are the sparse or dense points in continuous vector space.
- Learn “visual vocabulary” from these features, which is a clustering process.
- Quantize features using visual vocabulary.
- Represent images by frequencies of “visual words”, which estimates categorical distribution.

In previous studies, a number of intensity or edge-based visual feature extraction methods based on BoF [17] have been proposed and achieved good performance, including SIFT [19], GIST [1], etc. They are widely known as low-level image representations.

Despite of the great breakthrough of the BoF-based models, the limitations of them obviously exist. Firstly, traditional BoF-based models represent an image as an order-less collection of local features and dismisses all information about the spatial layout of features. To explore spatial information, Lazebnik et al. [20] proposed an effective structural object descriptor based on the BoF-based features, namely Spatial Pyramid Matching (SPM). The SPM model consists of two key components: pyramid matching kernels [21] and spatial matching scheme. An illustration of a typical SPM model has been depicted in Figure 2.1. Under the scope of spatial pyramid scheme, some BoF-based features can be greatly improved with the incorporation of spatial information. For example, the popular fast dense SIFT descriptor: Pyramid Histogram Of visual Words (PHOW) [22] represents images based on the appearance descriptors with the image spatial layout. The PHOW features are extracted at multiple scales and quantized into M bins of visual words by the pyramid-based k-means clustering. As shown in Figure 2.2, a sequence of

FIGURE 2.2: An example of PHOW descriptors with two levels of pyramid, where $L = 2$.

levels of the pyramid on an image is constructed for the sake of spatial matching. The procedure of spatial matching scheme is illustrated as follows: Suppose that the pyramid level $l = 0, \dots, L$, and the grid at level l has 4^l cells. Multiplied by the size of visual words, level 0 has M bins of the histogram intersections, level 1 is represented by $4M$ bins and so on. The final concatenated histogram has a dimensionality of $M \sum_{l=0}^L 4^l$. After the spatial pyramid representation is formed, the image is matched by computing a three dimensional approximated histogram map for the pyramid matching kernel. Thus, each histogram descriptor is expanded into a vector of three dimensional outcome. Consequently, $3M \sum_{l=0}^L 4^l$ dimensional PHOW features are obtained in total. As an extension of SPM, Locality-constrained Linear Coding (LLC) method [23] in place of the hard vector quantization (VQ) in traditional SPM. In the following, effective sparse coding (SC) based SPM, and its variants such as LScSPM [24], LR-Sc⁺SPM [25], and CCLR-Sc⁺SPM [26] have received considerable attentions. Besides LLC encoding methods, the (improved) Fisher vector (FV) [27] encoding has been proposed by using a spatial pyramid scheme [28], which has won the champion in the ImageNet ILSVRC2012 [5] competition.

The second limitation of the BoF models is that there is a semantic gap between low-level feature representation and the meaning of the image content. To address this semantic gap, back in 2010, Li et al. proposed Object bank (OB) [29] as high-level image features using object filter banks, where an image is represented as a scale-invariant response map of a bunch of pre-trained generic object detectors

regardless of the different visual tasks. The OB representation was originally designed to be as the response of a “generalized object convolution”. By obtaining the objects from the popular datasets such as LabelMe [30], ImageNet [5] and Flickr online images, the authors of OB took the intersection set of the most frequent 1000 objects, resulting in 200 basic objects. This development is the preliminary design of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [31], which indeed boosts the breakthrough of deep learning in various computer vision tasks. As the development of many deep features, the object bank representation can be considered as the beginning of the deep learning revolution of the 2010s. Next, we discuss the deep visual feature representation learning methods, especially the state-of-the-art convolutional neural network baselines.

2.1.2 Deep Feature Representation

As one of the most hotly discussed representation learning methods, deep learning [32] explores multiple levels of representation, where higher-level feature representation describes more abstract aspects of the images. The transitional levels of representation are transformed from the lower-level representation using multiple layers of neural network in a hierarchical architecture. In the literature, deep learning architectures can be divided into three categories: supervised, semi-supervised and unsupervised learning. Among these categories, supervised learning approaches are the most practical and developed form of machine-learning, no matter deep or not. In particular, this thesis focuses on the aspect of image classification, which is one of the most common applications for supervised learning approaches. Deep neural network (DNN) is one regime for image classification under the scope of deep learning. Inspired by the biological neural systems, DNN mimics the mechanism of computation in human brain and performs as a type of feed-forward neural network with a large amount of computational units (i.e., neurons). In general, DNNs are trained with stochastic Gradient Descent (SGD) [33] and back-propagation (BP) algorithm [34]. As a prototypical example of advanced DNNs, convolutional neural network (CNN or ConvNet) was firstly proposed by Fukushima back in 1988 [35] that has not been greatly promoted until LeCun et al. [36] applied gradient-based learning algorithm in 1990s. Since the early 2000s, the phenomenal CNN have been widely applied with exceptional success to the

FIGURE 2.3: The overall structures of AlexNet with 5 convolutional (conv) layers and 3 fully connected (FC) layers.

FIGURE 2.4: The overall structures of Network In Network with the stacking of three multilayer perceptron convolutional (mlp_conv) layers and one global average pooling (GAP) layer.

detection [37, 38], segmentation [39, 40], recognition of objects [41, 42] and regions [43], and captioning [44, 45] in images.

A classical CNN, such as AlexNet [41], consists of convolutional layers wrapped with pooling steps and then followed by the fully connected layers as dense layers in a standard multilayer neural network. Figure 2.3 shows an AlexNet structure with 5 convolutional layers and 3 fully-connected layers. Due to the “data-hungry” characteristics, CNN are usually trained on large image datasets. However, learning of thousands of filters on large-scale datasets is computational expensive. For example, the 8-layer AlexNet model [41] employed 650,000 neurons in total and has 60 million parameters to train on ImageNet ILSVRC [5] dataset, which usually takes more than one week to train via backpropagation algorithms on the GPUs.

Beginning from AlexNet, the development of CNN architecture grows explosively. To address the expensive computational power issue, ZFNet [46] improved upon

FIGURE 2.5: An example diagram of GoogLeNet with “Inception” modules. The Inception modules combine convolutions of different sizes. Note that the multiple softmax at different inception modules are not shown due to the space.

AlexNet by reducing the network parameters of AlexNet. Network in Network (NiN) was the first CNN model that used multilayer perceptron (MLP) in the convolutional filters (1×1 filter) of the network. Furthermore, it used the global average pooling (GAP) algorithm as a replacement of fully connected layers to significantly reduce the network parameters and avoided the overfitting issue. Figure 2.4 shows an NiN structure with three MLP-convolutional layers and one GAP layer. The VGGNets [47, 48] were a series of AlexNet-based convolutional network, which show that the depth of the CNN architecture is an intractable factor to achieve better performance and generalization in classification.

FIGURE 2.6: An example residual block in the ResNet.

More recently, the GoogLeNet [49] has been proposed to incorporate the Inception layers to reduce computational complexity of the CNN architecture (22 layers in total). The residual network architecture ResNet [50] has been developed for the purpose of ultra-deep networks. Using the residual connections (including input to layer in addition to non-linear transform), ResNet can go further up to 1202 layers in a single end to end architecture. The core inception module and residual module of GoogLeNet and ResNet are exhibited in Figure 2.5 and Figure 2.6, respectively. Some newly proposed networks continuously improved the state-of-the-art classification performance, such as DenseNet [51], CapsuleNet [52], etc. In the following chapters, many of the aforementioned CNN baselines are implemented

FIGURE 2.7: The framework for CNN transferring and Fine-tuning using pre-trained VGG as example. FC_a is an adaptation layer and FC_b is the output layer of the new task.

as deep visual feature extractors in our proposed methods.

The well-trained CNN have good adaptability of parameters transferring, and are often fine-tuned when applied to a specific tasks. For instance, DeCAF [42], Caffe [53] and Oquab et al. [54] used the deep representations obtained from the pre-trained models and re-trained the adaptation layers for different tasks. The activations of the penultimate layer (prior to the softmax layer) of the CNN can be treated as powerful CNN feature vectors [55]. An example of CNN transferring and fine-tuning is shown in Figure 2.7. In this thesis, we transfer and fine-tune several aforementioned pre-trained CNN models to learn multimodal shared representation in the proposed methods.

FIGURE 2.8: A toy example on the one-hot encoding method.

2.2 Text Representation Learning

2.2.1 Bag-of-Words Representation

The bag-of-words (BoW) modeling is a classical text representation learning method for many natural language processing (NLP) tasks, including one-hot word encoding, term frequency/inverse document frequency (TF-IDF) [56] filtering and n -grams modeling. In essence, the bag-of-words model is a type of vector space model (VSM), in which a text (such as a sentence or a document) is utterly thrown into a bag of its words regardless of syntactic structure and word order.

In this section, we will take 1-of- m encoding (or one-hot encoding) method as a typical example of BoW-based modeling to vectorize the text. For example, as shown in Figure 2.8, given two sentences D_1 : “*The cup is on the table*” and D_2 : “*The mug is on the desk*” and the codebook of the word bag is defined as a set of the basis term: $\{“cup”, “table”, “mug”, “desk”\}$, the two sentences are mapped into a fix-length vector representation: $\mathbf{d} = [p_1, p_2, \dots, p_m]$, where p_i is the number of occurrences of i -th basis term in the given text document and m is the size of codebook of the word bag. Any word exceeding the codebook is automatically ignored, which are quantized as zero values in the vectors. As a result, D_1 and D_2 are represented by $\mathbf{d}_1 = [1, 1, 0, 0]$ and $\mathbf{d}_2 = [0, 0, 1, 1]$, respectively.

In practice, the BoW-based models are very popular in the application of building text features for sentence-level or document-level classification due to its ease of use. However, BoW-based model produces sparse and very high-dimensional vector representations. Based on the word occurrence statistics, the BoW-based models

can barely convey the word semantics underlying the text documents. It is because that the high dimensionality of BoW-based vector space often leads to the curse of dimensionality. To address the curse of dimensionality, we will briefly review a popular technique for dimensionality reduction: Latent Semantic Analysis (LSA) in the field text mining and NLP.

Latent Semantic Analysis: Latent semantic analysis (LSA) [57] is a classical dimensionality reduction technique, especially for the distributional semantics. The development of LSA is to analyze the relationships between a set of documents and the terms they contain. With the distributional hypothesis, LSA assumes that words that are semantically similar will occur in the similar text clusters. A matrix with word occurrence is firstly constructed by the BoW-based model, and then the singular value decomposition (SVD) [58] technique is adopted to reduce the dimensionality of BoW text features. Given a set of training data represented by the TF-IDF vectors: $\mathbf{X} \in \mathbb{R}^{N \times d}$, where N and d are the data sample size and the data dimensionality, respectively. We can simply transform the data $\mathbf{X}$ by mapping from the original space to a new space with lower dimensionality:

$$\mathbf{H} = \mathbf{XW} \quad (2.4)$$

where $\mathbf{W} \in \mathbb{R}^{d \times r}$ is the linear transformation matrix ($r < d$) and $\mathbf{H} \in \mathbb{R}^{N \times r}$ denotes the new data representation. To train the transformation matrix $\mathbf{W}$ in Eq. (2.4), the optimization solution for LSA can be derived as follows:

$$\mathbf{W}^* = \underset{\mathbf{W} \in \mathbb{R}^{d \times r}}{\operatorname{argmax}} \operatorname{Tr} \left(\frac{\mathbf{W}^T \mathbf{L} \mathbf{W}}{\mathbf{W}^T \mathbf{B} \mathbf{W}} \right) \quad (2.5)$$

where Tr denotes the trace of a matrix, $\mathbf{L}$ represents the covariance matrix of $\mathbf{X}$, which is the quantity need to be enhanced, and $\mathbf{B}$ denotes the quantity need to be suppress. In particular, matrices $\mathbf{L}$ and $\mathbf{B}$ in LSA can be calculated as:

$$\text{LSA} : \begin{cases} \mathbf{L} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^T \\ \mathbf{B} = \mathbf{I}_d \end{cases}, \quad (2.6)$$

where $\mathbf{x}^{(i)} \in \mathbb{R}^{d \times 1}$ is the TF-IDF vector of i -th sample, and N denotes the number of samples. The second term in $\mathbf{B}$ of Eq. (2.6) represents a regularization term, where $\mathbf{I}_d$ denotes an identity matrix with a size of $d \times d$. Based on Eq. (2.6), the

singular value decomposition (SVD) is performed on the data matrix $\mathbf{L} \in \mathbb{R}^{d \times d}$ to obtain a latent space spanned by r dominant eigenvectors corresponding to the r largest eigenvalues.

In fact, the LSA dimensionality reduction method is essentially the same as applying Principal Component Analysis (PCA) [59] method on the word occurrence matrix, except that PCA subtracts off the means of $\mathbf{x}^{(i)}$ in Eq. (2.6), which is written by:

$$\mathbf{L}_{pca} = \frac{1}{N} \sum_{i=1}^N (\mathbf{x}^{(i)} - \bar{\mathbf{x}})(\mathbf{x}^{(i)} - \bar{\mathbf{x}})^T \quad (2.7)$$

where $\bar{\mathbf{x}}$ represents the mean vector of the all samples. Although the SVD-based feature extraction methods can effectively address the high dimensionality issue of the BoW-based models, it has limitations in terms of discriminative power. For instance, the mapping of LSA usually follows the direction of maximum variance with respect to the eigenvalue ranking of the covariance matrix. This may lead to insufficient discriminative power in the lower-dimensional feature space in various NLP tasks. In addition, considering the computational complexity of solving Eq. (2.6) is $\mathcal{O}(d^3)$, it is not efficient to introduce dimensionality reduction approaches in text mining problems when the data dimensionality is very high (over 10,000). Thus, we implement TF-IDF method without feature dimensionality reduction in Chapter 3 for the ease of use.

Besides the curse of dimensionality issue, the BoW-based models involve exact word matching that accounts for word occurrence, which ignores the semantic correlation between words, and therefore lack semantic generalization. As shown in Figure 2.8, there exists no semantic correlation between the two BoW-based vector representations $\mathbf{d}_1$ and $\mathbf{d}_2$ if the word vector distance or similarity is measured, which leads to a discrepancy of their true hypernym-hyponym correlation in the semantic taxonomy. Hence, the BoW descriptors can hardly conserve the semantics of the text at a high level. Therefore, we explore higher level semantic text features by developing semantic matching methods between word embeddings in Chapter 4 and Chapter 5.

FIGURE 2.9: The framework for Neural Language Model (LM).

2.2.2 Word Embeddings

Word embeddings, also called distributed word representation, are often reported to produce very competitive performance in the era of deep learning for many NLP tasks, including sentence classification [2, 60] and matching [61], sentiment analysis [62], topic categorization [63, 64], etc. As opposed to the high-dimensional and sparse text representations obtained from BoW-based models, word embeddings are the distributed word vectors that encode semantic and syntactic correlation of words into low-dimensional and dense space. Some existing neural network models, such as *word2vec*¹ [65, 66] and *GloVe* [67] learned dense vector representations of words (embeddings) through unsupervised neural language modeling (LM) [68]. A typical framework of LM is shown in Figure 2.9.

The most popular model for word embeddings is a two-layer natural language model proposed by Mikov et al. [65]. A vector space where semantically similar words are assigned similar vectors is learned by this model. Two efficient architectures including Continuous Bag-of-Words (CBow) and Skip-gram are proposed in [65]. CBow attempted to predict a word given the context words, while Skip-gram predicted a window of words given a single word. The training objectives of CBow and Skip-gram are to maximize the log probability of any surrounding word given

¹available at <https://code.google.com/p/word2vec/>

the entire context and the current center word, respectively, which are as defined as follows:

$$\Gamma(\theta)_{CBoW} = \frac{1}{M} \sum_{m=1}^M \log p(s_m | s_{context(m)}) \quad (2.8)$$

$$\Gamma(\theta)_{skip} = \frac{1}{M} \sum_{m=1}^M \sum_{j \in context(m)} \log p(s_j | s_m) \quad (2.9)$$

where $context(m)$ denotes the list of words that close to s_m , and the probability $p(\cdot|\cdot)$ is defined as the softmax function of the dot product between embeddings of words. Considering its efficient and unsupervised training protocol, word embeddings can be learned from a large-scale unannotated corpus with limited computational resources. Using word embedding representations, the semantic similarity between words is preserved in the lower dimensional vector space. Therefore, word embeddings have the ability of characterizing high-level features with semantic meanings. In this way, raw text data can be mapped into a dense and real-valued vector space, so that the mid-level CNN-based image representation and text representation can be combined in feature level. Furthermore, good adaptability of pre-trained word embeddings means they are universal features extractors and can be used in various classification tasks.

2.2.3 CNN Representation

Although CNN are originally developed for computer vision tasks, the state-of-the-art text mining tasks also benefit from the merits of CNN, including sentence classification [2, 60] and matching [61], sentiment analysis [62], topic categorization [63, 64] and many others. We begin with the CNN model on the sparse one-hot encoding vectors [69]: given a short sentence $D = \text{“mat on cat”}$ with the vocabulary $V = \{\text{“cat”, “mat”, “on”, “sat”, “the”}\}$. Then, the document D is converted to a 1-of-5 vector: $\mathbf{d} = [1, 1, 1, 0, 0]$. Similar to the CNN on images, convolving filters sweep over the region of interest of $\mathbf{d}$, resulting in new features. Since BoW-CNN models are deemed to be quite robust to data sparsity, it is suitable for long-text analysis, such as document categorization [70].

The rapid development of CNN for NLP is attributed to the emerge of word embedding techniques. By utilizing the pre-trained word vectors as universal feature

FIGURE 2.10: An architecture of one convolutional layer CNN with word embedding input for text representation.

extractors, several CNN models [2, 63, 71] have produced the mid-level text feature representation by taking the convolution operation on the input word vectors, which is very powerful for short or incomplete text classification [62].

In the case of document representation, as shown in Figure 2.10, let $\mathbf{x}_i \in \mathbb{R}^d$ denote a d -dimensional vector representation (in embedding space) of the i -th word in the document, the document of length l is represented as:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_l] \in \mathbb{R}^{ld}. \quad (2.10)$$

Kim [2] proposed a simple but effective CNN with one layer of 1-D convolving window filters on the document matrix $\mathbf{X}$. Every convolving window filter $\mathbf{W}_x \in \mathbb{R}^{rd}$ swipes through the entire rows of $\mathbf{X}$ to generate a feature map, where the width of $\mathbf{W}_x$ is the same as the width of $\mathbf{x}_i$ and the window slides over r words in Eq. (2.10). The general form of the output a_i activated by a non-linear function is

TABLE 2.1: Common nonlinear activation functions in CNN. “tanh”: hyperbolic tangent; “ReLU”: rectified linear units.

Function	Equation
sigmoid	$f(x) = \frac{1}{1 + \exp^{-x}}$
tanh	$f(x) = \frac{2}{1 + \exp^{-2x}} - 1$
ReLU	$f(x) = \max(0, x)$

given by:

$$a_i = f\left(\sum_i^{i+r-1} \mathbf{W}_x \cdot \mathbf{x}_i + b\right), \quad (2.11)$$

where $f(\cdot)$ is a standard nonlinear activation (bounded) function such as hyperbolic tangent (tanh), and $\mathbf{W}_x, b$ are the parameters to be trained. The available and most common nonlinear activation functions are listed in Table 2.1. The feature map is produced by sliding the convolving window filter through the entire documents:

$$\mathbf{a} = [a_1, a_2, \dots, a_{l+r-1}]. \quad (2.12)$$

In the convolutional layer, multiple filters capture multiple feature maps as outputs, which allow different filters to act as individual feature extractors. Then the max pooling is performed over each feature map to extract the largest values from the map vector as shown in Eq. (2.12) (i.e., the most semantically salient features over feature maps). Finally, the preliminary features are concatenated in the penultimate layer (fully-connected layer). The weights of the network are commonly learned using the stochastic gradient descent (SGD) and the backpropagation algorithm.

Other popular text modeling methods, for example, recurrent neural network (RNN) [72] and its gated variants long short term memory (LSTM) [73], gated recurrent unit (GRU) [74] are also applicable to text representation learning and classification tasks, including sentiment analysis [75, 76], relation classification [77], etc. The architectures of RNN as well as GRU can be seen in Figure 2.11. Different from CNN-based methods, the RNN-based approaches emphasize the coherence of the web text and treats the text classification problem as a sequence-based classification problem. Recently, some advanced work such as [78] explored the neural network-based fusion of CNN image representation and bidirectional GRU text representation for object classification.

FIGURE 2.11: The architecture of RNN (with 3 time steps) encoder and GRU encoder.

2.3 Multimodal Data Fusion

In this section, we briefly introduce some baseline methods on multimodal data representation learning. Multiple modalities of data are usually obtained from different feature subsets about the common target data. Most intuitively, different modalities of features can be considered as different views on the target data. For example, a person can be identified by multiple views of characteristics, such as facial features, fingerprint, or signature. In fact, all these different views of description on this person are complementary to each other, making the person identification more comprehensive and accurate. On the other hand, since the semantic correlation between different features may underlie in cross-modal feature representation, the advanced and higher-level neural network frameworks can be exploited to discover the multimodal dependencies of different feature subsets.

2.3.1 Multi-View-Based Data Integration

According to [79], the multi-view learning methods are commonly divided into three categories: co-training, multi-kernel learning (MKL), and subspace learning. Among these methods, co-training [80] is one of the earliest approaches that relies on two assumptions: 1) the classification on each view is sufficient on its own. 2) the target functions in different views can share and predict common class labels for co-occurring features. Nigam et al. [81] proposed a co-training scheme using

FIGURE 2.12: Subspace learning framework for image-text data fusion.

expectation and maximization (EM) algorithm on each view, which outperforms original co-training method in many applications.

The subspace learning [82–84] in multi-view learning has been developed to draw insights from the cross-modal correspondence. The major goal of subspace learning-based methods is to obtain a latent subspace shared by multiple views given the assumption that the input views are created based on this latent subspace, as shown in Figure 2.12. In single-view learning approaches, principle component analysis (PCA) has been widely applied to obtain a latent subspace for single-view data. In the multi-view cases, canonical correlation analysis (CCA) [85] plays the similar role of PCA. The CCA and its kernelized version [86, 87] are used to learn a common space of CNN-based multimodal representations for cross-modal retrieval and classification [88]. In addition, heterogeneous transfer learning methods [82, 89, 90] typically learn a common latent space of different modalities, which transfer the heterogeneous knowledge to domains where very little training data is available. The commonality among these subspace learning methods is they all use similar images from the web as a medium between training images and web text data to train the latent subspace.

However, the multi-view-based approaches above somehow dismiss the independence of multiple modalities. By contrast, independent learning approaches, such as multi-kernel learning [3, 9, 91–93], build specific models for individual modality data and combine the models linearly or non-linearly for optimization. Our implementation in Chapter 3 is a representative approach of MKL learning for

decision-level data fusion that uses image modality data and text modality data to train separate models and then combines their results systematically.

2.3.2 Neural Network-based Data Integration

With the advent of deep neural networks (DNNs), advanced multimodal learning based on DNNs have been exploited in a wide range of applications. For example, multimodal deep Boltzmann machines [94] have been applied to image-text representation learning, multimodal recurrent neural networks (RNNs) methods have been proposed for image captioning [45, 95–99], multimodal CNN have been presented for image-sentence matching and visual question answering [100, 101], multimodal deep auto-encoder architectures have been proposed for speech classification and video hyper-linking [102, 103], and so on. Unexceptionally, these methods involve the joint feature representation that enables the DNN models to fill in missing modalities, when some modalities are less informative or absent. Moreover, the joint representation preserves the similarity of concepts for improved generalization. In Chapter 4 and Chapter 5 of this thesis, our proposed multimodal fusion model relies on the text-visual joint feature representation learning using the idea of feature level data integration.

2.4 Summary of Related Work

Here, we demonstrate the connection of our proposed pipelines elaborated in the following sections and their related work shown in this literature review section. Our discussion will follow the order of the subsections with respect to their corresponding related work in this thesis.

The BoF-based image representation methods shown in Section 2.1.1 are used as important image descriptors and carefully compared in the experiments section of Chapter 3. The deep image feature representations mentioned in Section 2.1.2 are widely used as the baseline image representations in our Chapter 4, Chapter 5, and Chapter 6.

The BoW-based text representation methods illustrated in Section 2.2.1 produce simple but effective low-level text representations in NLP, which have been widely

used and discussed in the context of Chapter 3-Chapter 6. The word embedding representations mentioned in Section 2.2.2 as the pre-trained word vectors have been applied as the representation of the input text for the neural networks proposed in Chapter 4-Chapter 6. Next, our semantic matching neural network in Chapter 4 and semantic convolutional networks in Chapter 5 both draws their motivation from the text-based CNN models reviewed in Section 2.2.3.

Our Chapter 3 uses image modality and text modality separately and then combines them in decision level, which is an exemplar of MKL learning for decision-level data fusion strategy. Therefore, it is highly related to the existing methods discussed in Section 2.3.1. Meanwhile, Section 2.3.2 shows the neural network-based data fusion methods, from which our feature-level fusion learning models in Chapter 4 and Chapter 5 draw inspiration.

Chapter 3

Web Text-Aided Image Classification Based on Adaptive Decision-Level Fusion

In this chapter, we demonstrate a decision-level multimodal fusion framework on image modality and its corresponding web text modality for aiding scene classification. Several classification tasks has been performed on scene-centric and object-centric datasets to show the effectiveness of our proposed framework.

The rest of the chapter is organized as follows. Section 3.1 presents the motivation of this work and the overview of the system. Section 3.2 elaborates the class label-based and bag-of-words (BoW)-based text representation learning methods for image classification. The adaptive multimodal fusion algorithm based on the web text-based and training image-based classification is given in Section 3.3. Experimental results and conclusion are presented in Section 3.4 and Section 3.5, respectively.

3.1 Introduction

3.1.1 Motivations

Suppose we are given two visually similar but semantically different scenes “bocce” and “croquet” to classify. As shown in Figure 3.1, the two scenes are ambiguous to machines since both scenes contain some identical objects: grass, balls, people, etc. Besides, the similar motion of the people in the images makes the scene classification even more difficult. In fact, identifying of some similar scenes like Figure 3.1 may not be a simple task for humans as well, especially for those who are not familiar with sport events. But humans are naturally conscious of how to leverage the previous learned knowledge and the external knowledge when they not knowing the answer to general problems [104].

We thereafter imitate the human logic by the way of fusion of training data and web knowledge in high level. The new framework trains an additional combiner in addition to the image-based classifier, which adaptively tradeoffs the decision scores obtained from the training image-based classifier and the web text-based classifier.

To train the web text-based classifier, we investigate the appropriate feature extraction methods based on the results obtained from the online search-based text retrieval. But a real challenge encountered in online search-based method is that the web resources are often noisy, and the returned texts of online search may provide inaccurate or even wrong information about the class of the query ¹ images. To avoid the negative impact brought by web text, we firstly extract class label directly from the texts of the returned images. This is the simplest and most intuitive attempt to focus on the most significant semantic information of the web text only. Sometimes, however, the returned text does not contain the class label of the image, and then the entire text will be discarded. To address this limitation, we intend to explore beneficial semantic information underlying the context other than image class labels. Specifically, we put the entire web text corpus into a BoW-based model to learn BoW-based text representation. Here, a noise filtering algorithm is necessarily developed to remove the irrelevant web texts for the BoW-based model.

¹Both training and test images can be considered as query images in our work.

FIGURE 3.1: Two similar but different scenes, “bocce” and “croquet”.

Since web resources have great reliability diversity, it may not be an optimal practice to allocate fixed weights to the visual feature-based and text feature-based classifiers as in [9–11, 105]. In this chapter, an adaptive fusion algorithm is developed for the integration of the visual feature-based and web textual feature-based classification results. Therefore, this adaptive fusion algorithm allows the combiner to change as the web resources reliability varies, which is inspired by the multisensory integration mechanism of human whose adaptability is achieved by reliability-dependent weighting of different sensory modalities[106].

As shown in Figure 3.2, our proposed framework consists of three components: training image-based classification, web text representation learning and classification, and an adaptive decision-level multimodal fusion. Two separate classifiers on these two modality data are learned respectively, which is followed by an adaptive combiner in decision level. This adaptive combiner is performed by applying pairwise adaptive weight vectors $\mathbf{w}_p$ and $\mathbf{w}_t$ to image-based and text-based classification scores. The goal of handling different modality of data separately is to reduce the vulnerable interaction of heterogeneous feature space.

3.2 Web Text Representation Learning

3.2.1 Online Web Text Retrieval and Data Preprocessing

The merits of applying online web text retrieval method have been discussed in Chapter 1. Here, we demonstrate the procedure of this retrieval method based on online reverse image search. Given a query image, we upload this image as a query to search visually similar images on the web using Google Image, and

FIGURE 3.2: An overview of the proposed framework with adaptive decision-level fusion algorithm.

some examples of the Google Image search results can be found in Figure 3.3. Note that, the web text data for training is retrieved here if the query image is a training image, while the web text data for inference is obtained if the query image is a test data. It can be seen that the visually similar images are sorted based on visual similarity. We therefore find the first n returned images, which are already annotated by the web pages or personal uploading, containing a large amount of informative semantic information that facilitates understanding the visual contents within the images. Accordingly, the associated image captions and descriptions of the first n returned images are retrieved as auxiliary data, while the visually similar images are not involved in our framework.

In terms of data preprocessing, these captions and descriptions of web images are firstly considered as raw text data, which is then converted to word strings. Based on these word strings, the lexicons are generated from the raw strings by lexical analysis, which is known as *tokenization*. In particular, the raw text is tokenized into a sequence of alphabetic and non-alphabetic characters. Next, we perform the *data cleaning* techniques such as predefined “stopping words” (pronouns, connectives, prepositions, etc) removal and encoded data (ASCII, Lation-2, UTF-8, etc)

FIGURE 3.3: An example of the procedure of online web text search (upper half) and data preprocessing (lower half) with the class label “climbing” when $n = 4$. The class label information in the image captions and descriptions is circled in red.

extraction. Finally, *word stemming* process is adopted here to remove the morphological affixes from words. For example, as shown in the Figure 3.3, “climbing” and “mountaineering” will be automatically transformed to their word root “climb” and “mountainer” after stemming, respectively. When extended to N query images, a collection of their associated web text data is obtained by online web text retrieval and processed by data preparation to yield N strings of useful documents. Next, we discuss the procedure of learning class label-based and BoW-based web text representations.

3.2.2 Class Label-Based Text Representation

It is observed that the semantic category, i.e., class label of each of the returned images, might be contained in the captions or descriptions of the images, as highlighted by the red cycles in Figure 3.3. The simplest text representation learning method is to extract the class label information directly from the web text data based on exact word matching. According to the class information given by specific tasks (e.g., with h image classes), this method assigns a h -dimensional vector to represent the web text data of a query image, which can be given as:

$$\mathbf{c} = [c_1, c_2, \dots, c_h] \quad (3.1)$$

where c_i is a binary number to denote the presence or absence of the i -th class label in the context and h is total number of classes in the task ($i \leq h$). Assuming that we given the web text of a query image: “*rock climbing in the peak district of Alps mountains*” and the three class labels of the task: {“*coast*”, “*highway*”, “*mountain*”}, the text data is then represented by a vector of $\mathbf{c} = [0, 0, 1]$ following Eq. (3.1).

3.2.3 BoW-Based Text Representation

We have demonstrated the feature extraction approach on class label of the web text data. Although the class label-based extraction method is very straightforward, it dismisses all other important context except class labels, such as “*climbing*” and “*Alps*” in the example given in Section 3.2.2. Moreover, despite the incorrect fourth returned result of Figure 3.3, we observe that the class label-based approach can hardly deal with the indicative information underlying the second returned result of Figure 3.3 as no class label information is found in the context, which is a kind of waste of information.

In this section, we address this problem by representing web text using Bag-of-Words (BoW) model. Instead of direct extraction of the class label information, we exploit the preprocessed raw text data to create a web text corpus and establish the BoW-based model. Before introducing the BoW-based model, a noise filtering method on the web returned resources is necessarily discussed in the following texts.

FIGURE 3.4: An overview of the VSM-based noise filtering.

Vector Space Model (VSM)-Based Noise Filtering: Exploring web resource automatically by machine itself is a very challenging task since web resources are often non-cooperative and noisy, for example, the returned images and texts might be irrelevant to the query image as shown in the last returned image and text in Figure 3.3. On one hand, we hope to increase the volume of the web texts so that the retrieved web text resource is sufficiently abundant. But on the other hand, the irrelevant text should be discarded since irrelevant texts provide irrelevant or even wrong information about the query image. Vector space model (VSM) [107] is a popular method for text representation and text similarity evaluation. To remove the task-unrelated text, we measure the cosine similarity between the web returned text data and the class label-related reference documents in a VSM model. As shown in Figure 3.4, the class labels can be used as text queries to obtain reference documents from the web. Suppose we are given h class labels, h reference documents are obtained using Wikipedia keyword-based retrieval with respect to the class labels. The raw web reference documents are then pre-processed by word stemming and “stopping words” removal (same as Section 3.2.1).

Given a query image, the online search-based retrieval method returns the web text data of n visually similar web images as a document of n strings. Accordingly, the document is vectorized into a collection of n sub-vectors $\mathbf{s} = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n]$, and the web reference documents are vectorized into a set of h sub-vectors $\mathbf{d} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_h]$ in VSM. Next, we evaluate the cosine similarity between each individual sub vector $\mathbf{s}_i$ and the entries vectors $\mathbf{d}_j$:

$$\cos \theta_j = \frac{\mathbf{s}_i \cdot \mathbf{d}_j}{\|\mathbf{s}_i\| \cdot \|\mathbf{d}_j\|}. \quad (3.2)$$

There are two scenarios for the cosine similarity: $\mathbf{s}_i$ and $\mathbf{d}_j$ are positively correlated, i.e., $\cos \theta_j > 0$ and negatively correlated, i.e., $\cos \theta_j < 0$. $\mathbf{s}_i$ is retained if and only if $\max \cos \theta_j \geq T_{cs}$, where T_{cs} is a pre-defined threshold. Otherwise, it will be eliminated from $\mathbf{s}$. As a result of the noise filtering, we obtain a more precise web text collection $\mathbf{s}'$ with n' sub-vectors for a test image, where $n' \leq n$. The overall text retrieval and noise filtering algorithm can be found in Algorithm 1.

Algorithm 1 BoW-Based Text Representation Learning

Input: h relevant reference document vectors $\mathbf{d}$, N query images.

Output: N TF-IDF weight vectors of the query images.

1. Upload each image as a query to search visually similar images on the web using Google Image.
 2. Return the first n image captions/descriptions of each target image $\mathbf{s} \in \mathbb{R}^n$.
 3. **for** $0 \leq i \leq n$ **do**
 - 1: **while** j **not exceed** h **do**
 - 2: Measure the cosine similarity $\cos \theta_j$ between $\mathbf{s}_i$ and $\mathbf{d}_j$.
 - 3: **end while**
 - 4: **if** $\max \cos \theta_j \leq T_{cs}$ **then**
 - 5: Remove $\mathbf{s}_i$ out of $\mathbf{s}$.
 - 6: **else**
 - 7: Retain $\mathbf{s}_i$ in $\mathbf{s}$.
 - 8: **end if**
 4. **end for**
 5. Extend the filter algorithm to N images and compute the TF-IDF weights of retained text documents.
-

BoW Feature Extraction: Using the noise filtering algorithm discussed above, we could effectively remove irrelevant noisy components from the raw data and build a new text corpus for representation learning use. In this section, the term frequency-inverse document frequency (TF-IDF) weighting scheme is adopted to learn low-level BoW features based on the text corpus. As a prototype of the BoW-based models, the TF-IDF algorithm offsets the number of times a word appears in the web text data of a query image by the frequency of the word in the corpus (i.e., the web text of all query images). As the name implies, the TF-IDF weights are the combination of term frequency and inverse document frequency, where the

logarithmically scaled term frequency is commonly used and can be written by:

$$tf(t, d) = 1 + \log f(t, d), \quad (3.3)$$

where $f(t, d)$ derives the number of times that term t presents in document d . The smoothed inverse document frequency is to determine if a term is common or rare across all the documents:

$$idf(t, d) = \log\left(1 + \frac{N}{n_t}\right), \quad (3.4)$$

where N denotes the number of documents in the corpus and n_t represents the number of documents that the term t occurs. The inverse document frequency also plays a role of filtering some words that appear more frequently in general. Hence, the TF-IDF features are the products of tf and idf weights, which reflect how important a word is to a document in a corpus. Consequently, the TF-IDF model maps the web text into a fix-length vector with respect to the vocabulary size of the corpus.

3.3 Adaptive Fusion based on Multimodal Image Classification

As shown in Figure. 3.2, we learn the image classifier based on the visual features of the training images in parallel to web text-based classifier. These two separate classifiers are used to learn heterogeneous data independently, and an adaptive decision-level fusion strategy is applied upon the multimodal classification results to boost the image classification performance.

3.3.1 Web Text-Based Classification

The class label-based and BoW-based text representations learned from Section 3.2 are used to train text classifiers, respectively. Considering the multi-class classification, we implement one-vs-rest (one-vs-all) linear SVM. In this approach, h binary classifiers are employed, each of which separates class j , $j = 1, \dots, h$ from the rest $h - 1$ classes. Once the SVM classifier has been trained, the text feature vectors

of a test image is delivered to h classifiers. The resulting image-to-label vector of a query image is denoted by $\mathbf{t} = [t_1, t_2, \dots, t_h]^T$, where t_j denotes the decision score of the testing (query) image belonging to the j -th class. The highest score $\max_{1 \leq j \leq h} \{t_j\}$ indexes to the class that the web document of a testing image belongs to. Here, the web text-based decision score vector $\mathbf{t}$ will be used for adaptive multimodal fusion, which is illustrated in Section 3.3.3.

3.3.2 Training Image-Based Classification

In the literature, a variety of visual feature extraction methods have been proposed. In particular, we investigate on the low-level feature descriptors, including GIST, dense SIFT (PHOW[22]), object bank [108], sparse coding Spatial Pyramid Matching (ScSPM [109]) and mid-level image representation learned from one of the VGG networks [48] in this chapter. GIST image descriptors are convolved with Gabor filters, where the returned feature maps are divided into grids to obtain the average sub-region feature values. PHOW features are extracted at multiple scales in multiple pyramid levels and quantized into the visual words using pyramid-based k-means clustering. Object bank, however, employs many pre-defined object detectors such as latent SVM object detectors to obtain the responses by three dimensional spatial pyramid mapping. Non-negative ScSPM trains a codebook based on extracted SIFT features. The sparse coding features are then calculated by feature pooling using spatial pyramid. We use the pre-trained VGG networks² as off-the-shelf feature extractors to capture deep visual features.

Then, the feature vectors learned from the aforementioned visual feature extraction methods are input to a classifier. Similar to the web text-based classifier, the outcome from the SVM classifier for image data is a h -dimensional vector $\mathbf{p} = [p_1, \dots, p_h]^T$, where p_j denotes the decision score of this testing sample belonging to the j -th class. Here, the classification results refer to the decision scores. In general, the highest score $\max_{1 \leq j \leq h} \{p_j\}$ reflects the correct class the testing instance belongs to.

²They are pre-trained on ImageNet ILSVRC [5] with 1000 object-centric classes.

3.3.3 Adaptive Multimodal Fusion

Motivated by the human logic of internal and external knowledge high-level aggregation, our proposed semantic meaning-level multimodal fusion framework is performed in three steps. First, an initial classification is performed by the pattern classifier learned from training images. Second, we search similar images on the Internet, and extract web text modality of visually similar web images. BoW-based web text classification will be made separately to improve the original performance in the image space. Last, a decision-based late fusion algorithm is performed. The fusion is actually done by applying an adaptive and trainable weight vector to pairwise concatenate the image-based and web text-based classification scores. The web text helps disambiguation, which in turn leads to more accurate and robust performance.

The adaptive fusion strategy is inspired by humans' superb adaptive capability. Research in neuron science, for example [106], has found that this adaptive capability is owing to the multisensory integration mechanism through divisive normalization. Thereby, we propose an adaptive fusion algorithm following human multisensory integration in decision level. In the adaptive fusion of the visual feature-based classifier and the textural feature-based classifier, the weights assigned to each class in each modality are adapted to each individual testing image based on the reliability of visual cues and textual cues of the testing image.

Multimodal fusion in decision level often employs a weighted linear summation model. For instance, the fused decision score of a testing image w.r.t the j -th class is given by:

$$s_j = w_{pj} \cdot p_j + w_{tj} \cdot t_j, \quad (3.5)$$

Different from the traditional multimodal fusion model that fixes the weights w_{pj} and w_{tj} for the visual feature-based and textual feature-based classifiers, we employ adaptive weights in this work. Weights w_{pj} and w_{tj} are now functions of coherence, normalized within the range 0 to 1. Given the decision scores of the visual feature-based and textual feature-based classification results, w_{pj} and w_{tj} are given by:

$$w_{pj} = \frac{1 + e^{\lambda_j \cdot p_j}}{1 + e^{\lambda_j \cdot (p_j + t_j)}} \quad (3.6)$$

and

$$w_{tj} = \frac{1 + e^{\lambda_j \cdot t_j}}{1 + e^{\lambda_j \cdot (p_j + t_j)}}, \quad (3.7)$$

where λ_j is a regularizer to produce reliability-dependent weights w_{pj} and w_{tj} , w.r.t the j -th class. Here, the regularizer λ_j is learned from the training data.

Assuming there are N training data from h classes. Different classes use different values of regularizers because the reliability of training data and web resources of different classes might be different. In K -fold cross validation [110], each of the N training data has been used once as a validation instance. When used as a validation, the decision scores of image i are denoted by $\mathbf{p}_i = [p_{i1}, \dots, p_{ih}]^T$ and $\mathbf{t}_i = [t_{i1}, \dots, t_{ih}]^T$ respectively. The fusion result is a linear combination of $\mathbf{t}_i$ and $\mathbf{p}_i$:

$$\mathbf{q}_i = [q_{i1}, \dots, q_{ih}]^T, \quad (3.8)$$

where

$$\begin{aligned} q_{ij} &= w_{pj} \cdot p_{ij} + w_{tj} \cdot t_{ij} \\ &= \frac{1 + e^{\lambda_j p_{ij}}}{1 + e^{\lambda_j (p_{ij} + t_{ij})}} \cdot p_{ij} + \frac{1 + e^{\lambda_j t_{ij}}}{1 + e^{\lambda_j (p_{ij} + t_{ij})}} \cdot t_{ij}. \end{aligned} \quad (3.9)$$

Given the class label vector $\mathbf{y}_i \in \mathbb{R}^h$ of the image i (i.e., a column vector with value 1 at the c^{th} position and 0 at other positions, and c is the class label of image i), we intend to make $\mathbf{q}_i$ be close to its target $\mathbf{y}_i$. In other words, the optimal regularizers $[\lambda_1, \dots, \lambda_h]^T$ can be obtained by minimizing the following cost function:

$$J = \frac{1}{N} \sum_{i=1}^N \|\mathbf{q}_i - \mathbf{y}_i\|^2. \quad (3.10)$$

Since this optimization to Eq. (3.10) is not a straightforward linear optimization problem, it cannot be solved using linear algorithms such as least squares method. Notably, the input data to the optimization function is h pairs decision scores obtained from the bimodal classifiers. Thus, we conduct h separate optimizations on Eq. (3.10), where the regularizers are derived separately.

Given a testing image, the classification outcomes from the training image-based classifiers and web resources are first obtained: $\mathbf{p} = [p_1, \dots, p_h]^T$, $\mathbf{t} = [t_1, \dots, t_h]^T$.

Algorithm 2 Adaptive Multimodal Fusion

Input: N labeled training image data, decision scores matrix $\mathbf{T}$ of the web text of training data, a testing image, decision scores vector $\mathbf{t}$ of the web text of testing data.

Output: The classification decision C of a testing image.

1. Conduct a K -fold cross validation on N training data.
2. **while** k **not exceed** K
 - 1: Perform classification on N/K training images to obtain the decision score matrix $\mathbf{P}^{(k)}$.
 - 2: Repeat computing for other $K - 1$ trails.
3. **end while**
4. Construct the linear sum $\mathbf{Q}$ of $\mathbf{P}$ and $\mathbf{T}$ by Eq. (3.9) and evaluate Eq. (3.10).
5. Solve nonlinear optimization problem to obtain the regularizers.
6. Perform classification on the testing images to obtain the decision score vector $\mathbf{p}$.
7. Fuse $\mathbf{p}$ and $\mathbf{t}$ and derive the decision score vector $\mathbf{q}$ by applying adaptive weights.
8. The final classification decision F is made by the highest score in $\mathbf{q}$ with Eq. (3.12).

The decision scores are then merged using the pairwise adaptive weights:

$$\mathbf{q} = [w_{p1}p_1 + w_{t1}t_1, \dots, w_{ph}p_h + w_{th}t_h]^T \quad (3.11)$$

The classification decision is finally made by the highest decision score in $\mathbf{q}$

$$F = \arg \max_{1 \leq j \leq h} q_j \quad (3.12)$$

The above fusion procedure is summarized in Algorithm 2.

3.4 Experiments and Results

In this section, we assess our image data and web resources fusion model for image classification using four benchmark datasets: UIUC-Sports dataset [111], Scene-15 dataset [20], Caltech-256 [112] dataset, and Pascal VOC-2012 dataset [113]. The

statistics of these four datasets as well as their auxiliary web text data are given in Table 3.1.

- **UIUC-Sports:** The UIUC-Sports dataset [111] contains 8 sports event categories: “rockclimbing” (194 images), “badminton” (200 images), “bocce” (137 images), “croquet” (236 images), “polo” (182 images), “rowing” (250 images), “sailing” (190 images), and “snowboarding” (190 images). The image number in each class ranges from 137 to 250, and there are 1579 images in total. Some example images in the dataset can be found in Figure 3.5. Note that we use the same experimental settings as [29]: randomly select 70 images from each class for training and 60 images for testing.
- **Scene-15:** The Scene-15 dataset [20] contains 15 natural and cultural landscape categories: “coast”, “forest”, “highway”, “inside_city”, “mountain”, “open_country”, “street”, “tall_building”, “office”, “bedroom”, “kitchen”, “living_room”, “industry”, “store”, “suburb”. Most images in the categories are gray-scale images. Figure 3.6 shows some example images in the dataset. The image number in each class varies from 210 to 410, and there are 4485 images in total. We randomly select 100 images from each class for training and another 100 images for testing as in [20].
- **Caltech-256:** The Caltech-256 [112], as an extension of Caltech-101 [114], is a challenging set of 256 object categories containing from 80 to 827 images per category. The mean number of images per category is 119. We follow the settings in [48], which randomly select 15 and 60 images per category for training and use the remaining images as test data.
- **Pascal VOC-2012:** The PASCAL VOC-2012 [113] classification dataset has over 11,500 training/validation³ images from 20 object categories: “aeroplane”, “bicycle”, “bird”, “bus”, “boat”, “bottle”, “car”, “cat”, “cow”, “chair”, “diningtable”, “dog”, “horse”, “motorbike”, “pottedplant”, “person”, “sofa”, “sheep”, “train”, “tvmonitor”. Due to the unavailability of the ground truth labels of VOC-2012 test data, We train the classification models on the training/validation images and upload the results of the test set to the challenge server for evaluation.

³This dataset has originally been split into training set, validation set, and test set by the challenge.

TABLE 3.1: Statistics of datasets (note that we show statistics of VOC12 validation set only due to the unavailability of test labels). N : Number of images. $|V|$: Vocabulary size. $|V_{abs}|$: Size of vocabulary that absent in the pre-trained word vectors.

Data	N	# categories per image		vocabulary size	
		$=1$	≥ 1	$ V $	$ V_{abs} $
UIUC-Sports	1579	1579	-	5192	547
Scene-15	4485	4485	-	10459	1292
Caltech-256	29780	29780	-	30236	3171
VOC12 (val)	11540	7567	3973	15051	1779

3.4.1 Compared Approaches

As stated above, we implement the multi-class linear SVM on visual and textual feature space using LIBLINEAR [115]. The methods based on the single image modality and multimodal fusion methods with the following visual features are compared in this chapter:

- **GIST descriptors** [1]: The images are resized into 256×256 ; the number of filters is set to 4; the number of orientations per scale is set to [8 8 8 8].
- **PHOW descriptors** [22]: We perform the dense SIFT descriptors via VLFeat toolbox [116]. Images are resized to 640×480 pixels; regular grids spacing is set to [4 6 8 10]; SIFT descriptors of 6×6 pixels; pyramid level $L=3$; visual vocabulary size $M=200$.
- **Object bank descriptors** [29]: We implement the object bank code in [108] with 200 object detectors; the number of detection scales is set to 12; pyramid level $L=3$.
- **ScSPM descriptors** [109]: We implement the ScSPM representation using the code in [109]: images are resized to 256×256 ; grids spacing is 6; the size of patch is 16; pyramid level $L=3$; the number of iteration is set to 50.
- **VGG deep representation** [48]: We adopt the activations of the penultimate layer of the VGG network, which contains 16 layers of computation

units trained on ILSVRC-2012 dataset involving 1000 ImageNet classes. Before feeding into this pre-trained network, we rescale the images into fixed size of 224×224 .

3.4.2 Experimental Setups and Evaluation Metrics

We capture the web text data of the top 50 most visually similar returned web images via Google inverse search using Selenium automating browser⁴. The text analytic is implemented by the natural language toolkit (NLTK) [117] and Scikit machine-learning toolkit [118]. We set empirical value $T_{cs} = 0.2$ in the VSM-based noise filtering algorithm as illustrated in Section 3.2.3. One may concern about the information overlap emerges between the training and testing data in web data retrieval. To address this concern, we evaluate the impact of information overlap by the experimental statistics. For instance, we find 41 out of 10,991 testing images in the VOC-2012 dataset involving overlapping information with that of the training data, which confirms the occurrence of information overlap. However, the overlap ratio is approximately 0.4%, which is an extremely insignificant to the experimental results of our proposed methods.

Since the training and test sets of UIUC-Sports, Scene-15, and Caltech-256 datasets are randomly split, we run the experiments on these three datasets for 30 trails to achieve statistical meaningful results and all the random splits were kept static for all compared methods. The mean and standard deviation of the top-1 accuracy (**mAcc**) are given as $\tilde{d}$ and S_d , respectively. Then the paired t -test is calculated by:

$$t = \frac{\tilde{d}}{S_d/\sqrt{n}}, \quad (3.13)$$

where $n = 30$. Under the null hypothesis, this statistic follows a t -distribution with $n - 1 = 29$ degrees of freedom. We observe that p -values of three paired t -tests are all less than 0.01 from the table of t -distribution, indicating that the performance improvements achieved in Table 3.2, Table 3.4, and Table 3.5 (see the following result sections) are significantly meaningful.

The evaluation metric of VOC-2012 is slightly different from that of the aforementioned three datasets. According to the benchmark baselines in the literature,

⁴available at <https://www.seleniumhq.org/>

FIGURE 3.5: Example images for the UIUC-Sports dataset.

TABLE 3.2: Mean accuracy (%) of the state-of-the-art feature extractors (average over 30 trials) on UIUC-Sports.

Algorithm	Visual feature	Label Fusion	BoW Fusion
GIST [1]	64.2 ± 1.7	77.4 ± 1.9	85.8 ± 1.2
ScSPM [109]	80.3 ± 0.9	85.9 ± 0.9	91.0 ± 1.1
OB [29]	77.9 ± 0.9	87.0 ± 1.0	88.2 ± 1.2
PHOW [22]	83.9 ± 1.1	88.2 ± 1.3	91.1 ± 1.0
VGG [48]	95.7 ± 1.2	94.9 ± 2.3	95.6 ± 0.9

the performance on the VOC-2012 is evaluated by the average precision (**AP**) and mean average precision (**mAP**).

3.4.3 Results of UIUC-Sports dataset

We firstly validate the effectiveness of our proposed adaptive decision-level fusion models on the UIUC-Sports dataset. In Table 3.2, we compare the classification results of the state-of-the-art feature descriptors upon the UIUC-Sports dataset when they are used alone or combined with web resources under the multimodal fusion framework. Obviously, significant improvements are achieved in all the 4 low-level feature extraction methods. Without using any pre-trained deep networks, the benchmark result of the UIUC-Sports dataset was achieved by our BoW-based multimodal fusion method with an overall accuracy of $91.11 \pm 1.04\%$. Since the deep VGG16 features are obtained and trained based on large-scale ImageNet database, the classification performance on VGG16 features surpasses that on other low-level features by a large margin. However, we observe that fusing with either class label information or BoW representation of the web text cannot help to boost the performance of single modality classification on UIUC-Sports dataset.

TABLE 3.3: Per-class accuracy (%) comparison of the model using visual descriptors only, text descriptors only, and the proposed decision-level fusion models on UIUC-Sports.(average over 30 trials).

Method	rockc.	badmi.	bocce	croqu.	polo	rowin.	sailin.	snowb.	mAcc
PHOW [22]	93.1	91.6	63.5	76.1	84.7	88.2	90.7	81.3	83.9
Class-label	86.6	64.9	66.2	77.1	78.2	84.3	87.9	80.4	78.2
BoW	92.8	81.3	75.4	79.7	84.3	85.4	88.1	81.9	83.5
Label Fusion	97.2	87.9	66.4	76.8	90.7	96	98.3	89.9	88.2
BoW Fusion	98.3	95	76.1	83.3	90.6	95.5	99.4	90.1	91.1

In order to investigate the effectiveness of using web text for image classification, we also show the experimental results of using web text alone on the UIUC-Sports dataset. For example, Table 3.3 shows the classification performance for each class when PHOW descriptors are used without fusion, with class label-based fusion, and with BoW-based fusion, respectively. Besides that, we also compare with the per-class classification accuracy of the class label method and BoW-based method. In the comparison of using single modality data, it is interesting to observe that both two text-based methods outperform the PHOW-based method in classes “bocce” and “croquet”, suggesting the complementarity of web text data and local image data. Furthermore, we can see adding external web text information leads to significant improvement in results. Although the class label-based fusion method has already achieved substantial improvements in overall, the accuracy of class “badminton” slightly drops. By contrast, the BoW-based fusion method achieves considerable improvements in all classes as shown in the Tabel 3.3. Among these categories, the result of class “bocce” is improved by 13%, which is a much higher improvement than all previous reported works on UIUC-Sports dataset.

3.4.4 Results of Scene-15 dataset

In this section, we implement our framework on another scene-related dataset Scene-15. The classification results of the state-of-the-art feature descriptors with and without the multimodal fusion are given in Tabel 3.4. The classification accuracy of class label-based method and BoW-based method using web text alone are $68.9 \pm 1.1\%$ and $70.1 \pm 1.0\%$, respectively. Regardless of the deep VGG features, the BoW-based multimodal fusion method with ScSPM descriptors achieves the best result of $86.29 \pm 0.62\%$, while the class label-based multimodal fusion method slightly improves the overall accuracy. Similar to our observation in Table 3.2, the

FIGURE 3.6: Example images for the Scene-15 dataset. The starred 8 categories are the originally obtained from [1] (a.k.a., LabelMe8 dataset).

TABLE 3.4: Classification accuracy (%) of the state-of-the-art feature extractors (average over 30 trials) on Scene-15 Dataset.

Algorithm	Visual feature	Label Fusion	BoW Fusion
GIST [1]	72.6 ± 1.2	76.4 ± 1.8	81.5 ± 1.0
ScSPM [109]	80.3 ± 1.1	82.7 ± 1.3	86.3 ± 0.6
OB [29]	80.2 ± 0.8	83.1 ± 0.9	86.0 ± 1.3
PHOW [22]	81.4 ± 1.2	82.8 ± 1.5	84.2 ± 0.7
VGG [48]	90.2 ± 0.5	90.2 ± 1.6	90.7 ± 1.3

TABLE 3.5: Classification accuracy (%) on the Caltech-256 dataset. The number of “15” and “60” in the table denotes the number of data for training.

Method	Visual feature		Label Fusion		BoW Fusion	
	15	60	15	60	15	60
GIST [1]	19.7	31.5	28.4	38.7	31.9	41.3
ScSPM [109]	27.7	40.1	33.2	45.8	34.1	47.2
PHOW [22]	28.3	42.8	32.9	47.6	33.1	49.3
VGG [48]	70.9	76.9	71.0	76.9	72.5	78.8

performance bottleneck when using VGG16 features in the multimodal fusion is also reflected in our experiments on Scene-15 dataset.

By comparing the experimental results of two scene-centric datasets vertically, we find that the performance of class label-based fusion model is more sensitive to data, while the new method produces more robust performance in the general situations of scene classification. This reveals that the semantic information underlying the text is indeed meaningful for the understanding of the image contents.

3.4.5 Results of Caltech-256 dataset

In order to verify our proposed fusion framework makes substantial contribution to image classification, we extend it to object-centric datasets: Caltech-256 and VOC-2012 (the latter results are shown in Section 3.4.6). The classification results of the state-of-the-art feature descriptors with and without the multimodal fusion are given in Tabel 3.5. Note that the generation of object filter bank in [29] uses Caltech-101/256 object categories of images to train the object detectors, which is not suitable for the evaluation on Caltech datasets due to the training/test overlap. In particular, we observe that either the class label-based fusion or BoW-based fusion outperforms the the GIST, ScSPM, PHOW descriptors as expected. Yet, the classification performance improvements seem to be relatively unsatisfactory

in the challenging Caltech-256 task. Moreover, we can see an apparent gain in accuracy with the number of training samples increases (from 15 to 60). By contrast, the two text models using class label and BoW information bring about more semantic knowledge from different view of the text modality. The classification accuracy of these two models for 60 training samples per class are 57.8% and 63.4%, respectively, which overwhelm all the shallow image descriptors of local image data by a large margin. The performance bottleneck is finally alleviated until the deep image descriptors are adopted, such as VGG16. Furthermore, our proposed fusion frameworks can slightly boost the performance for deep visual features as well. In overall, the experiments on Caltech-256 have proven the effectiveness of the text-aided fusion frameworks in the object classification problem.

3.4.6 Results on VOC-2012 dataset

The classification on VOC-2012 dataset is a multi-label multi-class classification task, which is very difficult to tackle due to the overlapping of foreground and background objects within the images. For example, the “chair” image in Figure 3.7 is also labeled as a “person” and “table” image in the dataset because the image has all these three objects simultaneously, and we can not differentiate which one is the foreground object.

Table 3.6 gives the results of VOC-2012 for image classification. For the classification results based on single image modality data, we observe that the classification performance on low-level visual features is not comparable to that on the VGG deep visual features. Interestingly, the two text models using web text alone also outperform the low-level visual features by large margins, which indicates the effectiveness of the web text. Thus, we can investigate the multimodal fusion models for image classification with web aided text data. Table 3.6 also shows that the VGG model benefits from the fusion of text knowledge that class label information and BoW representation brought in. Compared to the results of the method using single image modality VGG features, the two proposed decision-level fusion methods improve over the classification mAP by 1.1% and 1.2% respectively. Moreover, the per-class AP shows that our proposed two fusion methods combined can improve as many as 16 categories of the VOC-2012 test set in total. Therefore, it proves the

FIGURE 3.7: Example images for the VOC-2012 dataset.

practical importance of introducing web text data in decision-level fusion models for multi-label object classification.

TABLE 3.6: Per-class AP (%) of the decision-level fusion methods on VOC-2012 dataset. Note that the fusion methods shown in the last two rows of the table are wrapped with VGG16 visual features.

Method	plane	bike	bird	boat	bt1	bus	car	cat	chair	cow
PHOW [22]	65.6	33.5	41.8	43.0	6.7	12.8	33.1	42.9	9.9	13.4
Class-label	65.2	55.0	49.4	46.1	28.8	62.1	59.5	59.8	42.4	41.9
BoW	91.1	64.7	83.9	76.5	33.9	68.9	61.8	90.2	44.8	71.2
VGG [48]	97.9	85.5	94.6	89.6	59.6	92.4	77.8	95.9	69.0	81.1
Label Fusion	97.8	82.6	94.7	94.9	60.4	91.5	81.1	96.2	70.5	81.4
BoW Fusion	98.1	81.3	95.2	96.5	62.0	91.7	78.9	94.8	71.2	83.1

Method	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	mAP
PHOW [22]	7.5	34.8	18.5	22.7	59.9	12.5	35.2	22.6	51.8	40.0	42.1
Class-label	29.8	55.9	51.4	62.7	73.1	39.1	49.7	29.3	60.0	20.5	50.9
BoW	47.1	88.7	82.9	68.7	78.0	36.1	74.3	38.9	79.6	55.1	66.8
VGG [48]	60.3	95.1	89.9	88.6	94.7	52.1	85.8	61.1	96.1	80.8	82.5
Label Fusion	74.3	93.3	91.2	90.8	92.1	52.2	85.7	65.8	95.7	79.2	83.6
BoW Fusion	69.3	93.3	94.5	89.9	90.4	52.1	86.2	66.0	96.3	82.1	83.7

3.5 Conclusions

In this chapter, we have proposed an adaptive multimodal fusion framework in decision level that uses both training data and web resources for scene classification. The class label-based and BoW-based text representations are explored for multimodal fusion in this work. Experimental results on the benchmark datasets show that the proposed text-aided image classification framework could significantly improve classification performance. Experimental results also show that the adaptive multimodal fusion mechanism can effectively individualize reliability-dependent weighting modulation for every new observation. The advantage of our adaptive decision-level fusion lies in the capacity of further boosting the classification performance of using visual features only after the classifier training is done

Chapter 4

Web Text-Aided Image Classification by Incorporating Task-Specific Semantic Matching Neural Network

In this chapter, we propose a novel semantic matching neural network (SMNN) for web text representation learning to reveal more meaningful semantics. The main objective of our proposed SMNN model is to learn high-level text features that have improved reliability and applicability, compared to the text features obtained from other methods discussed in Chapter 3. As opposed to the naive class label-based text representation and sparse 1-of- V BoW-based vectors, the SMNN model encodes semantic and syntactic information of the web text data into dense and real-valued text representation through a word-wise semantic matching algorithm. The importance of this development is that it allows us to explore the semantic correlation underlying the image modality and web text modality in a multimodal joint feature representation. In particular, this exploration is referred to as a neural network (NN)-based fusion framework in feature level for web text-aided image classification.

4.1 Introduction

In this section, we demonstrate the motivations behind our proposed methodology and the related text filtering methods inspiring our work.

4.1.1 Motivations and Overview

Our previous decision-level fusion framework deals with the multimodal data from multiple views. The visual features extracted from images and text features extracted from web text are used separately to train respective classifiers. The results are later fused at the semantic high-level through linear combination of the classifiers, where the weights of the classifiers are adaptive to different reliabilities of visual and text features. This adaptive decision-level multimodal fusion system preserves the information of multimodal structure, whereas it omits the underlying correlations across multiple modalities. To some extent, this limitation is mainly caused by the hindrance of different representation space of the heterogeneous data. In other words, we should be able to explore the semantic correlation from disparate modality data if the high-level semantic representation is available to incorporate.

For this purpose, we attempt to develop high-level semantic representation based on the word embedding technique, which utilizes dense and real-valued numerical vectors to represent semantics of words. In particular, we develop the semantic matching neural network (SMNN) model to extract high-level discriminative text features. Then, the CNN visual features from images and SMNN text features from auxiliary web text are concatenated and trained using a unified neural network, i.e., feature-level fusion framework. In this feature-level fusion framework, we assume that the image representation and the text representation are mutually complementary.

Figure 4.1 shows the general procedure and the proposed architecture for feature-level fusion model using SMNN for image classification. For a test image with web text information, we learn the final prediction by following the forward pass of the trained neural network. As shown in Figure 4.1, a test image is processed by two modules. On one hand, the test image is directly passed to the pre-trained CNN model to extract visual features from the raw pixels. On the other hand, the image

FIGURE 4.1: An overview of the proposed web aided neural network-based fusion framework. Here the online web retrieval process of a test image is shown, and we do likewise for training images to extract web text data.

is also used as a query for online web retrieval [13], which is referred to as image-to-text web retrieval. We use the SMNN and pre-trained CNN models as feature extractors in this framework to build up the shared bimodal representation, which is followed by a multi-layer perceptron (MLP) learning to yield the final decision. In addition, the challenge of applying web retrieval is also obvious due to the notoriously noisy text data on the web. We should be able to cope with the web noise, which is suppressed by the max pooling operation in the proposed SMNN.

4.1.2 Text Filtering Methods

The keystone of the SMNN model is the employment of embedded text filters (i.e., semantic filters) and the semantic matching process quantified by cosine similarity measures between embedded text input and semantic filters.

In the past decade, the idea of text filtering has been applied extensively in text representation, from bag-of-words (BoW)-based models to the more recent deep neural networks, such as convolutional neural networks (CNN). On one hand, the traditional text filtering methods such as [119, 120] employ exact word matching (i.e., one-hot encoding) or weighted exact word matching (e.g., counted one-hot and TF-IDF). By contrast, the filtering operation of our SMNN is essentially a semantic matching process based on word-wise correlation quantified by the cosine similarity measure between word embeddings. Thus, the filtering operation produces a real-valued and dense representation rather than a sparse representation for the text data retrieved such as image annotations, which are usually very short. On the

other hand, the convolving text filters used in CNN such as [2, 71] are usually learned by backpropagation algorithm through iterative minimization of the loss function. As a non-iterative learning process, the training phase of SMNN is mainly the selection of words whose embeddings are used as the filters in the filtering layer of the SMNN. Once selected, the filter weights of the SMNN are fixed without requiring additional training. Furthermore, the filters learned by CNN are mid-level feature extractors without semantic meanings, whereas SMNN filters are the embeddings of words relevant to the class labels and hence the features produced by the SMNN are semantically more meaningful. Once extracted from the SMNN, the semantic text features are aggregated with the CNN visual features to form a unified representation of an image for classification. Experimental results show that fusion of visual features and text features achieves significant improvements over visual features alone for image classification.

4.2 Semantic Neural Network

In this section, we firstly introduce the basic idea of the SMNN architecture, and then elaborate the generation procedure of semantic filters. A brief comparison of CNN and SMNN is discussed as well.

4.2.1 Network Architecture

Our proposed semantic matching neural network (SMNN) is a single-filter-layer feed-forward network, which draws its inspiration from the simple sentence-level CNN [2] structurally. As shown in Figure 4.2, the input to the SMNN input layer is a vector formed by concatenating embeddings of words (word vectors) in the text data, similar to CNN [2]. The input layer of both CNN and SMNN is followed by a filtering layer consisting of multiple filters, where the filter width is the same as the dimensionality of word vectors. The global max pooling operation is applied at the pooling layer in both CNN and SMNN. However, there is a fundamental difference between SMNN and CNN. Each filter at the filtering layer of the SMNN corresponds to an English word in the word vector space, and hence the features extracted by SMNN are semantically meaningful.

FIGURE 4.2: Illustration of a Semantic Neural Network architecture for text classification: semantic-filter layer and pooling layer. The boxes (in different colors) of the semantic-filter layer represent different separate feature maps (sets of filter responses). The output text feature representation is finally fed to the classifier.

The filters of the semantic-filter layer shown in Figure 4.2 are the word vectors (embeddings) learned by a neural language model such as the word2vec model. Assuming $\mathbf{z}_i \in \mathbb{R}^d$ denotes the d -dimensional (e.g. $d = 300$) word vector of the i^{th} word in the web text data with l words, the text data can be represented by a ld -dimensional vector:

$$\mathcal{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_l] \in \mathbb{R}^{ld}, \quad (4.1)$$

This representation of the text data is applied to the input layer of SMNN.

Then, a filtering operation by filter $\mathbf{w} \in \mathbb{R}^d$ is successively applied to $\mathcal{Z}$ to extract a feature map (i.e., a set of filter responses). Unlike the operation of convolving window filters in the CNN neuron, the neuron in our SMNN measures the cosine similarity between $\mathbf{z}_i$ and $\mathbf{w}$. The output of a neuron in the semantic-filter layer is given by:

$$c_i = \frac{\mathbf{w} \cdot \mathbf{z}_i}{\|\mathbf{w}\| \cdot \|\mathbf{z}_i\|}. \quad (4.2)$$

The cosine similarity measure Eq. (4.2) is bounded between -1 and 1, which is similar to the activation function $f(\cdot)$ in Eq. (2.11). The value of c_i indicates the semantic closeness between $\mathbf{z}_i$ and filter $\mathbf{w}$. The filter $\mathbf{w}$ is sequentially applied to

$\mathcal{Z}$ to produce the following feature map:

$$\mathbf{c} = [c_1, c_2, \dots, c_l], \quad (4.3)$$

where c_i is depicted as a colored dot inside its corresponding box in the semantic-filter layer shown in Figure 4.2. Next, a max pooling is applied over the population of semantic feature map $\mathbf{c}$ to obtain the largest value $\hat{c} = \max\{\mathbf{c}\}$. The max pooling operation captures the most salient feature (highest similarity score) of $\mathcal{Z}$ with respect to filter $\mathbf{w}$. Notably, max pooling rather than other pooling methods such as average pooling is used in the SMNN because max pooling **suppresses** the noisy components of web text data by ignoring the irrelevant contexts with less semantic relatedness with the filters. Each filter, combined with the max pooling operation, produces one feature of the input $\mathcal{Z}$. The m filters of SMNN in the filtering layer produce a m -dimensional text feature vector $[\hat{c}_1, \hat{c}_2, \dots, \hat{c}_m]$. Figure 4.3 shows the t-SNE visualization [121] of the web text data crawled for image samples of the UIUC-Sports dataset (please refer to Table 3.1 for details of the data), represented by three different text representation methods, including BoW (TF-IDF) model [56], CNN model [2], and SMNN model. It is observed that the samples of the same class are tightly clustered in the SMNN feature space, which is beneficial to classification tasks. The text features will be fused with visual features obtained from the pre-trained CNN and input to a pattern classifier, which is discussed in Section 4.2.4.

4.2.2 Generation of Task-Specific Semantic Filters

The features generated by CNN are commonly considered as mid-level features since the convolving window filters may not capture the semantics within the text. In this chapter, we come up with *semantic filters* for high-level text representation. A semantic filter refers to a filter that corresponds to an English word in the word vector space and is semantically relevant to the class labels. In a sense, the semantic filters in the SMNN capture the semantics of the text in a similar manner that the object filters re-figure the object patterns [108].

Unlike the convolving window filters that are learned on training data through the backpropagation algorithm, the semantic filters are automatically selected from the world wide web in this thesis. The generation of semantic filters constitutes the

FIGURE 4.3: T-SNE visualization of web text data of UIUC-Sports. The web text data is represented by three different text features: (a) BoW (TF-IDF) feature, (b) CNN feature [2] (wrapped with word embeddings), and (c) our proposed SMNN feature. The class labels of data samples are denoted in the color bar on the right.

main training process of SMNN. We employ the web reference documents as prior knowledge, from which the representative words that best describe the image classes are automatically selected as filter words. The benefit of generating semantic filters based on web resources is that the filter vocabulary is maximally enriched.

Class labels are used as text queries to obtain reference documents from the web. Suppose we are given h class labels, h reference documents are obtained using Wikipedia keyword-based retrieval. The raw web reference documents are then pre-processed by stemming and “stopping words” (pronouns, connectives, prepositions, etc) removal. The filter word selection procedure is illustrated in Figure 4.4. Firstly, the word-wise semantic correlations between class-label queries and their corresponding web reference documents are quantified by Wu-Palmer (WP) similarity measure [122]. This is different from [120] and [119], where exact word matching was used. Wu-Palmer (WP) similarity measure evaluates semantic correlation based on the hyponymy similarity in virtue of WordNet¹. Given a class-label query in the form of synset s_q and a word from its corresponding web reference document in the form of synset s_o , the Wu-Palmer (WP) similarity between s_q and

¹is a large lexical database for English, which groups nouns, verbs, adjectives and adverbs into sets of cognitive synonyms (i.e., synsets).

FIGURE 4.4: Flowchart of the semantic filter bank formulation. Semantic filters are word embeddings of prior knowledge-based representative words that best describes the image classes.

s_o is evaluated in terms of their least common subsumer (LCS) as follows:

$$WP(s_q, s_o) = \frac{2 \cdot d(LCS)}{d(s_q) + d(s_o)}, \quad (4.4)$$

where $d(\cdot)$ denotes the taxonomic distance between the root of taxonomy and a certain synset. Because a WordNet synset contains multiple words, $WP(\cdot, \cdot)$ is actually a set of similarity scores valued between 0 and 1. We take the average of the WP similarity scores. By setting a hard threshold T (e.g., $T = 0.5$) on the averaged WP similarity score with respect to every class-label query and its corresponding web reference document, we obtain m words that are highly relevant to the class labels. Finally, the selected m words are mapped into the vector space, via for example word2vec, and are used as filters in the filtering layer. The suitable value of m is application-specific (see Section 4.3.4).

4.2.3 Difference Between SMNN and CNN

Although SMNN is architecturally similar to the CNN that has a single convolutional layer, there exists fundamental difference between the convolving window filters in CNN and the semantic filters in SMNN. The convolving window filters of CNN have no semantic meanings, and the filters are learned from training data iteratively by backpropagation algorithm. By contrast, the semantic filters of SMNN are selected from open resources on the web. In addition, the semantic filters of SMNN do not need additional training once they are selected (see Section 4.2.2).

In fact, the part prior to the last layer of CNN and SMNN can be considered as neural network-based feature extractor. CNN extracts salient features by minimizing multi-class logistic regression loss, and hence it includes the softmax classifier in the last layer throughout the backpropagation-based training process. By contrast, the feature extractor in SMNN attempts to maximize the discriminative power or class separability, without the need of including a classifier in the loop. SMNN is inspired by the insight that features will be highly discriminative if embeddings of words that characterize the classes are employed as feature descriptors. Similar ideas are used in the training of other types of neural networks such as radial basis function (RBF) neural network [123, 124]. The rationale can be explained by the following example. As shown in Figure 4.1, the text attached to the web image is: *2015 World rowing tournament champion*. Applying this short text to SMNN, saliently high output values are obtained at semantic filters corresponding to watercraft-specific words such as “boat”, “kayak”, “canoe”, etc, while relatively low output values are obtained at other filters such as the filter of words “ball”, “badminton”, “racket”, etc. This suggests that the semantic filters are capable of increasing the intra-class correlation and inter-class separation simultaneously, which in turn leads to good classification performance.

4.2.4 Visual-Text Feature-Level Fusion

As shown in Figure 4.1, the multimodal fusion is built upon aggregation of image and text representations via a unified neural network. The term “multimodal” here refers to the fact that both text and visual features are used for image classification. Firstly, we obtain the CNN visual features from the penultimate layer of a pre-trained CNN model following the method in [54]. Secondly, the text features are

extracted from the SMNN. The SMNN text features and the CNN visual features are finally concatenated to form a unified representation of the image.

We train a two-layer neural network including a hidden layer and a softmax output layer over the unified representation. Let the unified representation of an image be a vector $\mathbf{u} \in \mathbb{R}^I$, and the class-label be a h -dimensional categorical vector. The weights in the hidden layer and the softmax output layer are trained by minimizing the multi-class logistic regression loss (a.k.a cross-entropy loss). In forward propagation for output unit y , we have:

$$y = \text{softmax}(\theta_2 \cdot \tanh(\theta_1 \cdot \mathbf{u} + b_1) + b_2), \quad (4.5)$$

where $\theta_1 \in \mathbb{R}^{h \times v}$, $\theta_2 \in \mathbb{R}^{v \times I}$, and b_1, b_2 are the bias terms. We employ standard nonlinear activation function $\tanh$ and softmax for the hidden layer and softmax output layer, respectively. In the backward propagation, the weights of both layers are tuned using stochastic gradient descent optimization with mini batch scheme.

4.3 Experiments

In this section, we will firstly introduce the datasets and implementation details. Then, the classification results are shown from the perspective of single-label and multi-label scenarios, respectively. In addition, the generation results of semantic filters discussed in Section 4.2.2 are visualized in the end.

4.3.1 Experiment Setup

In order to verify the effectiveness of our proposed SMNN text features and web text aided image classification framework, we conduct experiments and compare the performance with other state-of-the-art methods on the same four datasets as the experimental implementations of Chapter 3, including: UIUC-Sports [111], Scene-15 [20], Caltech-256 [112] and Pascal VOC-2012 [113]. Recall that the details and statistics of these four datasets have been illustrated in Section 3.4 of Chapter 3. The external web text data of these four image datasets is obtained by online web retrieval method proposed in Chapter 3. To handle the noise and ambiguity that the external web text may bring about, we preprocess the raw text data and clean

the non-sense components by the “stopping words” (pronouns, digits, contractions, etc) removal. Moreover, we remove the low document frequency words (less than 5 presences) from the raw text that are suspicious of irrelevant information. All the characters are converted to lowercase for unity. We skip the process of word stemming and lemmatization because we use word embeddings to represent the external web text.

In the experiments, we obtain the deep features on the whole images from the penultimate layers of the publicly accessible VGG16/19 models [48], ResNet152 model [50], and Xception model [125]. Note that the above four types of deep features were all pre-trained based on ILSVRC-2012 dataset with 1000 ImageNet classes, which are implemented using Keras². The visual features of Caltech-256 and VOC-2012 datasets are computed at multiple scales following the settings in [48] and [126], respectively. We present the results of other state-of-the-arts methods using the ones reported in the original papers.

In terms of the training configuration of the proposed semantic neural network (SMNN), hyperparameters of SMNN for text representation are defined as follows. For the input layer, we employ the GoogleNews-based 300-dimensional word2vec embeddings that are used in the CNN [2] for fair comparison. Hyperbolic tangent (tanh) as non-linear activation function. Global max pooling as the max-over-time pooling. RMSprop as the optimizer for mini-batch gradient decent with learning rate of 0.001. The mini-batch size is set to 50. The dropout noise with rate of 0.5 is also utilized to avoid over-fitting. Softmax classifier is used in the neural network-based fusion network. We conduct all the experiments on four NVIDIA GTX 1080 GPUs with 8GB memory³.

For UIUC-Sports, Scene-15 and Caltech-256 datasets, we randomly split the data into the training/test sets, and report the mean classification accuracy (**mAcc**) over 20 runs. The challenging VOC-2012 dataset is a multi-label classification task, where at least one ground truth label is tagged to every image. The evaluation metric of VOC-2012 is slightly different from that of UIUC-Sports, Scene-15, and Caltech-256. According to the benchmark baselines in the literature, the performance on the VOC-2012 is evaluated by the mean average precision (**mAP**).

²availale at <https://keras.io/> with Tensorflow

³We use the same GPU-based machine to run the experiments in Chapter 5 and Chapter 6 as well.

TABLE 4.1: Classification mAcc results (%) with different text features on the auxiliary web text data of UIUC-Sports, Scene-15 and Caltech-256 datasets. The shown results on Caltech-256 are obtained using the 60 images/category setting.

Method	UIUC-Sports	Scene-15	Caltech-256
BoW	83.5 ± 2.5	70.1 ± 1.0	63.4 ± 1.5
CNN _{static} [2]	86.1 ± 1.5	72.7 ± 0.8	64.9 ± 1.9
SMNN	90.4 ± 2.2	75.3 ± 2.1	66.5 ± 3.4

In the following subsections, we show the experimental results and analysis of our proposed SMNN text features and web text aided image classification framework on four benchmark datasets. Firstly, we test the SMNN for text feature extraction and compare it with two typical text representation methods including BoW [56] and CNN_{static} [2]. Secondly, our proposed SMNN+visual CNN composition for web text aided image classification method is evaluated and compared with a few visual feature-based image classification including low-level visual features, deep visual features, and other baseline fusion methods of multiple features. In addition, some results of the semantic filters generated based on the web reference documents are also presented.

4.3.2 Results: Single-Label Classification

We first evaluate the effectiveness of our proposed SMNN text features, by comparing with other text features for classification upon the auxiliary web text data of UIUC-Sports, Scene-15 and Caltech-256 datasets. The experiments on UIUC-Sports and Scene-15 datasets aim to prove the effectiveness of our proposed SMNN text feature and the web text aided neural network-based fusion models on small scale datasets. Evaluation on the Caltech-256 dataset is to extend our proposed approaches to the larger dataset for object recognition problem. To make fair comparison, we apply one hidden layer and softmax classifier on different text features. The results are shown in Table 4.1. The better performance of the CNN_{static} method over the BoW approach is attributed to the discriminative power that semantic relatedness brings about. More importantly, experimental results show that semantically salient text features are generated by the semantic filtering algorithm of SMNN. Consequently, our proposed SMNN method achieves the best classification performance among three text representation methods. To validate the significance of the improvements achieved by our proposed SMNN method on

TABLE 4.2: Comparison of Top-1 mAcc (%) between our methods and other state-of-the-art methods on UIUC-Sports, Scene-15, and Caltech-256 datasets.

The starred methods show the results reported in Chapter 3.

Method	UIUC-Sports	Scene-15	Caltech-256	
			15	60
PHOW [13]	83.9	81.4	28.3	42.8
ScSPM [13]	80.3	80.3	27.7	40.1
LLC [23]	84.6	81.6	32.7	46.1
DDSFL [127]	86.9	86.9	-	-
IFV [27]	90.3	89.2	34.7	47.9
5-layer ConvNet [128]	-	-	65.0	74.2
VGG16 [48]	95.7	90.2	66.2	74.6
VGG19 [48]	96.1	91.1	72.0	77.8
ResNet152 [50]	96.9	92.4	68.7	78.2
Xception [125]	97.2	92.5	73.4	80.6
DeConvNet [129]	-	-	67.0	77.9
Deep SIFT+LLC [130]	-	-	34.6	42.8
ISPR+IFV [131]	92.1	91.1	-	-
CNN fusion [132]	94.8	90.7	-	-
CNN-sNBNL [133]	95.3	92.9	-	73.7
DDSFL+Caffe [127]	96.8	92.8	-	-
AMF (w/ PHOW)*	91.1	84.2	33.1	49.3
AMF (w/ VGG16)*	95.6	90.7	72.5	78.8
CNN _{static} +VGG16 [3]	96.4	92.7	67.2	77.1
SMNN+VGG16	98.0	95.0	73.3	80.3
SMNN+VGG19	98.1	95.3	74.3	81.9
SMNN+ResNet152	98.6	96.1	72.8	82.3
SMNN+Xception	98.9	96.5	75.3	84.7

the three single-label datasets, we conduct paired t -test to evaluate the mean and standard deviation difference between SMNN method and CNN_{static} method. Here we compare with CNN_{static} approach only because BoW method is less competitive according to the results shown in Table 4.1. For each dataset, we perform random training/test splits for 20 runs and all the random splits were kept static for all compared methods. The mean and standard deviation difference are denoted as $\tilde{d}$ and S_d , respectively. Then the paired t -test is calculated by the Eq. (3.13) shown in Chapter 3. Under the null hypothesis, this statistic follows a t -distribution with $n - 1 = 19$ degrees of freedom. We observe that p -values of three paired t -test are all less than 0.01 from the table of t -distribution, indicating that the better performance achieved by our proposed SMNN methods on the auxiliary web text data is significantly meaningful.

Table 4.2 shows the classification results of various visual feature extraction methods, and some multimodal fusion models of visual and text features. According to the diversity of modality of multiple features, we categorize the baseline methods into three types and compare our proposed method with: 1) Unimodal single visual features: shallow representations, such as dense SIFT (PHOW [105]), locality-constrained linear coding (LLC [23]), improved fisher vector [131], etc., and deep representations including VGG features [48], classic 5-layer ConvNet [128], and De-ConvNet [129]; 2) Unimodal multiple visual features fusion: Deep SIFT+LLC [130], ISPR+IFV [131], CNN fusion [132] and so on; 3) Multimodal multiple features fusion, including adaptive multimodal fusions (AMF⁴) and CNN_{static}+VGG16 [3].

In the first (upper) part of Table 4.2, we show the classification results that use visual features alone. It is observed that the deep visual features are more powerful image representations than various shallow representations on all three datasets due to the enormous volume of the ImageNet repository for training. When it comes specifically to Caltech-256, sufficient training samples result in further gain in accuracy with the increasing number of training samples. The second (middle) part of Table 4.2 compares the baseline methods that fuse different visual features. In general, fusion strategy with multiple visual features can boost the performance, especially for those involving deep visual features.

Next, we compare the performance of SMNN+visual CNN fusion framework with other methods that fuse visual and text features. The contrastive experiments aim to validate the better performance of the proposed approaches over other multimodal representations using the same external text data from the same auxiliary web source. The performance of the AMF (w/ VGG16) method indicates that the concatenations of CNN visual feature and BoW text feature achieves very good results at the decision level. While CNN_{static}+VGG16 model outperforms AMF method, our SMNN+VGG models further improve the classification performance. And the results of SMNN+ResNet152 model and SMNN+Xception model surpass that of SMNN+VGG models with more powerful image representations.

TABLE 4.3: Per-class AP (%) comparison between our proposed SMNN methods and other text features methods on the auxiliary web text data of VOC-2012 dataset.

Method	plane	bike	bird	boat	btl	bus	car	cat	chair	cow
BoW [56]	91.1	64.7	83.9	76.5	33.9	68.9	61.8	90.2	44.8	71.2
CNN _{static} [2]	93.3	65.9	84.4	78.6	35.1	74.5	63.1	91.8	46.5	70.8
SMNN	94.9	74.8	86.1	81.1	48.7	83.2	70.1	88.3	57.8	69.4

Method	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	mAP
BoW [56]	47.1	88.7	82.9	68.7	78.0	36.1	74.3	38.9	79.6	55.1	66.8
CNN _{static} [2]	54.4	88.5	86.4	73.8	79.3	34.5	71.9	36.9	86.5	57.3	68.7
SMNN	39.9	85.4	87.3	76.8	84.7	41.2	77.6	41.9	84.2	61.9	71.8

4.3.3 Results: Multi-Label Classification

In this section, we extend our experiments to a more complex and difficult scenario: multi-label image classification. Unlike the single-label problems discussed above, Pascal VOC-2012 challenges may have more than one objects from different classes within the images. To validate the effectiveness of our proposed SMNN features as well as the web aided neural network-based fusion approach on multi-label scenario, we compare the experimental results with the baseline methods selected from Table 4.1 and Table 4.2.

Table 4.3 and 4.4 show the class-wise average precision (AP) as well as the overall mean AP (mAP) of the listed methods. From the results in Table 4.3, we can see that the SMNN text features outperform CNN-static and BoW text features in 15 out of 20 per-class average precision results. As shown in Table 4.4, we split the results of baseline methods into two parts and compare our proposed multimodal fusion approaches with: 1) Visual features: DDSFL [127], HCP1000 [134], VGG features [48], and etc.; 2) Multimodal representation: CNN_{static}+VGG16 and adaptive multimodal fusion (AMFs) method shown in Chapter 3. Among the models using single modality features, the CNN-based models achieve better mAP performance than DDSFL methods with a large margin of per-class AP values. The re-implemented four pre-trained CNN models (i.e., VGG16, VGG19, ResNet152, and Xception models) outperform the baseline methods [54] and [134]. Moreover, the mAP of these four models are consistently improved by at most 3.4% when

⁴This method is referred to as the one shown in Chapter 3.

TABLE 4.4: Per-class AP (%) comparison between our proposed web aided neural network-based fusion methods and the baseline results on VOC-2012 dataset.

The starred method demonstrates the results reported in Chapter 3.

Method	plane	bike	bird	boat	btl	bus	car	cat	chair	cow
DDSFL [127]	77.7	42.5	45.4	53.3	24.0	72.2	50.6	54.2	47.4	26.0
DDSFL+Caffe	92.7	75.4	85.1	79.4	42.3	88.1	68.5	87.1	62.9	65.8
Oquab et al. [54]	93.5	78.4	87.7	80.9	57.3	85.0	81.6	89.4	66.9	73.8
HCP1000 [134]	97.7	83.0	93.2	87.2	59.6	88.2	81.9	94.7	66.9	81.6
VGG16 [48]	97.9	85.5	94.6	89.6	59.6	92.4	77.8	95.9	69.0	81.1
VGG19 [48]	97.5	83.9	94.3	88.3	60.7	92.6	77.9	96.2	68.9	82.8
ResNet152 [50]	98.3	87.5	96.5	91.0	67.2	92.8	79.1	97.5	72.8	87.3
Xception [125]	99.1	88.4	96.3	91.3	69.3	94.1	79.9	98.5	72.9	92.5
CNN+VGG16	98.1	83.4	92.7	90.1	58.4	89.5	81.1	94.2	68.5	81.3
AMF (w/ VGG16)*	98.1	81.3	95.2	96.5	62.0	91.7	78.9	94.8	71.2	83.1
SMNN+VGG16	98.7	90.2	95.9	92.5	64.5	94.2	80.1	97.5	71.7	90.2
SMNN+VGG19	98.6	89.4	95.5	91.5	64.4	93.4	81.9	97.1	73.6	90.1
SMNN+ResNet	99.1	90.1	97.2	92.5	67.7	94.6	81.0	98.6	74.5	91.6
SMNN+Xception	99.3	91.6	97.3	93.6	72.7	95.2	82.0	99.0	75.4	96.1

Method	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	mAP
DDSFL [127]	36.6	42.6	38.9	50.7	70.3	23.4	36.2	36.4	61.6	49.0	46.9
DDSFL+Caffe	60.6	80.2	74.0	76.4	92.3	43.6	74.3	53.2	88.4	73.7	73.2
Oquab et al. [54]	62.0	89.5	83.2	87.6	95.8	61.4	79.0	54.3	88.0	78.3	78.7
HCP1000 [134]	68.0	93.0	88.2	87.7	92.7	59.0	85.1	55.4	93.0	77.2	81.7
VGG16 [48]	60.3	95.1	89.9	88.6	94.76	52.1	85.8	61.1	96.1	80.8	82.5
VGG19 [48]	64.8	95.1	90.7	88.8	94.8	51.4	87.1	62.2	96.1	81.5	82.7
ResNet152 [50]	69.4	95.8	91.9	89.7	95.6	58.5	91.4	64.8	96.3	84.9	85.4
Xception [125]	72.2	96.9	97.4	91.5	93.9	54.5	95.6	66.5	97.1	87.5	86.7
CNN+VGG16	74.3	93.3	91.2	90.8	91.1	52.2	85.2	65.8	95.7	79.2	82.9
AMF (w/ VGG16)*	69.3	93.3	94.5	89.9	90.4	52.1	86.2	66.0	96.3	82.1	83.7
SMNN+VGG16	67.9	96.2	94.8	90.9	95.1	56.1	92.5	65.2	96.3	83.9	85.7
SMNN+VGG19	72.3	96.6	95.4	91.2	95.2	56.1	92.6	66.9	96.3	84.4	86.1
SMNN+ResNet	70.4	97.1	95.3	91.6	95.7	60.0	94.5	67.3	96.5	86.1	87.0
SMNN+Xception	75.4	98.1	98.0	93.0	94.4	57.5	97.3	70.9	97.8	88.5	88.7

they are combined with SMNN text features under our proposed web aided fusion framework. The SMNN+Xception model achieves the highest overall mAP of 88.7% among different multimodal representation learning approaches, which is very close to the state-of-the-art results in the literature on VOC-2012 classification task. The robust improvements shown in Table 4.4 indicate that the proposed SMNN features on the auxiliary web data can further boost the state-of-the-art performance on VOC-2012 when better image representations become available for use in the multimodal framework.

4.3.4 Results: Semantic Filters

To retrieve web text using online search-based method (i.e., Google reverse image search), we capture the auxiliary web text data of the top 50 most visually similar returned web images. The text is processed using the natural language toolkit (NLTK) [117]⁵ and Scikit machine learning toolkit [118]⁶. For SMNN, we generate the semantic filters using the method illustrated in Section 4.2.2, where the WP similarity threshold T is set to 0.5. Some selected filter words are listed in Table 4.5. The exemplary filter words for Caltech-256 are not shown here due to high dimensionality. The semantic filter windows of unigram with $m=185, 263, 378, 1055$ feature maps are obtained for UIUC-Sports, Scene-15, Pascal VOC-2012 and Caltech-256 datasets respectively.

The number of semantic filters selected fully depends on the data diversity of different datasets. The output of the filters in SMNN denotes the semantic correlation between the input text and every semantic filter. Therefore, the higher value of the filtering output is, the higher semantic correlation it has with respect to the semantic filter, and vice versa.

4.4 Conclusions

In this chapter, we have presented a novel learning method, called *semantic matching neural network* (SMNN) for extraction of semantic text features. The experimental results on the benchmark datasets have proved the effectiveness of the high-level SMNN text features. Moreover, we have presented an web text aided feature-level fusion framework in this work that validates the complementary effect of CNN-based image representation and SMNN-based text representation to improve image classification. Together the conclusion drawn in Chapter 3, this observation indicates that the semantic correlation underlying heterogeneous data is likewise crucial for better understanding of image content. Yet, the task-specific property of the semantic filters is a potential limitation of the SMNN model in terms of the applicability of this representation learning method, which will be addressed in next chapter.

⁵<http://www.nltk.org/>

⁶<http://scikit-learn.org/stable/>

TABLE 4.5: Examples of the some semantic filters for three datasets. We list the class labels of three dataset in the second column, which are the semantic queries to obtain the web reference documents. The filter names shown in the third column are selected from the preliminary vocabulary without their synonyms obtained from WordNet.

Data	Class label	Selected semantic filters
UIUC-Sports	rockclimbing, bocce, badminton, croquet, rowing, sailing, polo, snowboarding	climb, climber, route, fall, rock, abseil, top, climbers, rope, climbing, lead, net, hit, court, side, mountaineering, badminton, shuttlecock, racket, bocce, ball, hoops, balls, croquet, association, side, endeavour, hoop, golf, fishing, game, striker, scored, mallet, team, goal, horse, polo, sculling, race, water, oars, boats, crew, kayak, rower, sculling, rowing, force, wind, hull, sailing, vessel, forward, mast, yacht, sailor, sailboat, snowboard, snowboarding, snow, tricks, mountain, wall, peak, ski, skier, skiing
Scene-15	coast, forest, highway, city, mountain, country, street, tall building, office, store, living room, kitchen, bedroom, suburb, industrial	wave, water, surfing, wind, coast, sea, sand, trawler, fishing, swimming, rainforest, forest, plant, roots, coniferous, tree, wood, bark, trunk, leaves, highways, toll, vehicles, car, congestion, traffic, transport, road, way, lanes, apartment, temple, town, building, square, house, tower, cathedral, church, plaza, skiing, mountain, mount, snow, lake, river, reserve, adventure, park, canyon, street, city, urban, downtown, avenue, top, skyscraper, tallest, manhattan, grasslands, convenience stores, suburb, outskirts, master bedroom, living room, liquor store, kitchen, village, bathrooms, suburbia, grocery, headquarters, lounge, edifice
VOC-2012	aeroplane, bicycle, bird, bus, boat, bottle, car, cat, cow, chair, table, dog, tv, horse, motorbike, plant, person, sofa, sheep, train,	airplane, takeoff, airline, boeing, wing, helicopters, jet, aircraft, engine, aviation, bicycle, tires, racing, handlebars, suspension, wheel, gear, frame, cycling, pedals, feather, nest, bird, water, prey, beak, eggs, school, buses, trucks, coach, transport, vehicle, kayak, boat, canoe, hull, sailing, vessel, aluminium, glass, bottle, wine, drink, plastic, container, beverage, milk, beer, company, car, chevrolet, automobile, auto, lamborghini, motors, volkswagen, sports, ford, cat, pet chair, seat, bench, farms, milk, dairy, cow, beef, animals, table, dining, dog, horse, male, colt, scooter, motorcycle, cruiser, plant, child, mother, musician, human, old, people, party, actor, father, grandparent, clothing, girl, footwear, sportswear, fashion, accessory, makeup, cosmetics, headgear, celebrity, family, portrait, statue, furniture, room, couch, sheep, lamb, goat, subway, metro, transit, train, railway, stations, monitors, television, computer, display, screen

Chapter 5

Task-Generic Representation Learning Method Based on Semantic Convolutional Neural Network for Web Text-Aided Image Classification

The previous Chapter 4 introduced semantic matching neural network (SMNN) for semantic representation learning and feature-level fusion using task-specific semantic filters. This chapter further investigates a novel semantic convolutional neural network (s-CNN) model for semantic text representation learning, in which the semantic correlation is encoded based on task-generic semantic filters. In order to achieve good coverage of universal tasks in a certain problem (such as, object classification), surplus filters are inevitably used in the s-CNN network, which in turn leads to semantic overlaps and feature redundancy. To address this, we develop the so-called s-CNN Clustered (s-CNNC) models that uses filter clusters instead of individual filters. Interacting with the image CNN models, the s-CNNC models can further boost image classification under a multimodal framework, which can be trained end-to-end. In addition, we propose to use the external text information selectively in the multimodal CNN network to avoid the noise effect that neutralize the classification improvement.

This chapter is organized as follows. The background and related work of semantic feature learning are illustrated in Section 5.1. Section 5.2 elaborates the proposed s-CNN for external web text representation learning. Section 5.3 introduces the multimodal CNN-based data aggregation network for incremental image classification. Experiments and conclusion are given in Section 5.4 and Section 5.5, respectively.

5.1 Introduction

5.1.1 Motivations

There are two challenges need to be addressed when external web text is used for incremental image classification. The first challenge is the representation learning of web images' annotations is usually based on very short text, which focus more on the lexical correlation than syntactic correspondence. The second challenge is that the text data on the web is notoriously noisy, which can easily nullify the benefits that the external data brings about. To address the first challenge, we developed the semantic matching neural network (SMNN) in Chapter 4, which involves task-specific text filters in the neural network. In this chapter, we propose a task-generic variant of SMNN under a convolutional network architecture. In particular, we apply a semantic convolutional layer on the top of the external web text data represented by the word2vec-based word vectors [66] and develop semantic CNN (s-CNN) models. To deal with the second challenge, i.e., the noise effect introduced by external text information, we develop a noise reduction scheme when the text representation learned by the s-CNN models is combined with the image representation learned by the image CNN models. The motivation is to selectively use external text information so as to alleviate the negative impacts of the noisy web data.

Next, we incorporate image data and web text through mapping their CNN-based representations into a common, multimodal embedding space. Here, we employ a fully coupled multimodal CNN (mm-CNN) architecture, in which the joint representations learned from image and web text data are integrated. In summary, the major contributions of the proposed approach for text-aided image classification are:

- We learn task-generic semantic filters in the convolutional layer of s-CNNs rather than task-specific filters. Thus, the proposed s-CNN models are able to extract task-generic features and learn transferable text representation, in a similar manner of image CNN baselines.
- We employ pre-trained word vectors to design the semantic filters in the convolutional layer, which acts like universal text feature extractors. Thus, the semantic correlation between the input and filter word vectors is encoded to the high-level representation.
- We cluster the semantic filters with an additional max/average-over-cluster pooling layer, and develop two clustered variants of s-CNN model namely s-CNNC_{max}/s-CNNC_{mean}. These models can effectively solve the feature redundancy problem caused by the semantic overlaps of the filters.
- We develop a noise reduction scheme for the mm-CNN model to alleviate the negative impact of the noisy components underlying the web text data.

5.1.2 Semantic Feature Learning

The previous Chapter 4 proposed a learning method so-called semantic matching neural network (SMNN) that explores semantic text features for image classification. The core contribution of the SMNN model is the employment of embedded text filters with semantic meanings. Applying embedded semantic filters on the top of pre-trained input word vectors, the semantic correlation among the input text words and filter words are captured by the cosine similarity measure. The semantic correlation can explicitly reflect the most discriminative and separable text features in the SMNN model. The major limitation of SMNN model lies on the task-specific text filters, which may not be very efficient for the classification tasks that have a great population of categories (e.g., the 256 categories of Caltech-256 challenge).

5.2 Semantic Convolutional Neural Network

When applied to NLP tasks, the convolutional neural network consists of layers of convolution filtering that involve exact word matching [63, 71]. The advanced

FIGURE 5.1: **System overview.** Our proposed method consists of two main modules: transferring and fine-tuning on the pre-trained CNN model and learning high-level text representation via s-CNN with filter clustering (framed in the shades). The aggregation of the two modules is performed by training a unified neural network. This figure is best viewed in color.

network with the idea of semantic matching has been recently developed to extract text features [2, 14], which takes the word-wise correlation into account.

The architecture of our s-CNN model with filter clusters is shown in the upper part of Figure 5.1. Note that we use the same online text retrieval method in this chapter as that in [14] and [3]. As depicted in Figure 5.1, we acquire the visually similar images via content-based image search and the similar images on the web are directly queried by a test image. Accordingly, the external web text associated with the visually similar images of query images is simply collected without downloading web images. In the training stage, we prepare the web text for training images in the same way and follow the general assumption of label transfer made for semi-supervised learning [135]. Thus, we transfer the category labels of the training images to the unlabeled text data for classification.

Let $\mathbf{x} \in \mathbb{R}^v$ be a v -dimensional word vector representing one word in the external web text, which is obtained from the word2vec model trained on 100 billion words of Google News dataset. This mapping process not only encodes semantics to the word-level numerical representation, but also suppresses the irrelevant word strings by automatically discarding them (set as zero-valued vectors) if they do not appear in the vocabulary of the pre-trained model. Thus, the entire sequence of text with length n is quantified as:

$$\mathbf{X} = [\mathbf{x}_1; \mathbf{x}_2; \dots; \mathbf{x}_n] \in \mathbb{R}^{nv}, \quad (5.1)$$

which is the input to the convolutional layer of the network. The conventional CNN for NLP perform the one-dimensional multi-gram convolution between the input sequence $\mathbf{X}$ and the convolving filters. Since the text attached to web images are usually very short, the s-CNN model simplifies the computation with unigram convolution regardless of the word orders. The semantic filter response c_j is derived by the dot product of a semantic filter with pre-trained vector $\mathbf{w} \in \mathbb{R}^v$ and every single word in $\mathbf{X}$:

$$c_j = \mathbf{w}^\top \cdot \mathbf{x}_j + b, 1 \leq j \leq n \quad (5.2)$$

where $\mathbf{x}_j$ denotes the vector of the j -th word, and b is a bias term. $\mathbf{w}$ is the weight vector of the pre-trained semantic filters, which are generated using the algorithm presented in the following section.

5.2.1 Generation of Task-Generic Semantic Filters

The concept of semantic filters in the dense embedding space was first proposed in [14], which are task-specific. In this chapter, we design generic semantic filters for the convolutional layer of CNN to achieve better adaptability for different tasks. Our inspiration is that we can leverage the labeling information from the large-scale databases off-the-shelf to capture task-generic text filters for different types of visual recognition problems, including the scene recognition and object recognition problems, respectively. In this work, we build two semantic filter banks based on the categories (i.e., category queries) of the ImageNet1000 and the Places365-Standard

datasets¹. Here, we only implement the basic categories of the ImageNet1000 and Places365-Standard because the fine-grained categories are usually too confined to the specific tasks. The basic categories of these two datasets produce two exhaustive lists of object-centric and scene-centric entities that are commonly seen in reality.

To achieve a good coverage of universal filter words for different tasks, a knowledge base with sufficient lexical information is required, from which the keywords that closely related to the querying category can be extracted. In particular, we deploy the Google News knowledge base involving 3 million vocabulary, on which the word2vec models [65, 66] are trained for word embedding. The rationales are summarized as follows. First, this pre-defined lexicon has great capacity of vocabulary within the English corpora, which is composed of more than 100 billion optional words and multi-words expressions². Second, since this lexicon is completely task-independent, it is able to provide open word candidates for a variety of problems, including object and scene recognition. Most importantly, it assures the presence of embedding representation for selected filter words. Next, the 3 million Google News vocabulary is screened by occurrence check on basis of WordNet, where the words and phrases absent in WordNet are automatically discarded. This process enforces homogeneity of the lexical space of Google News vocabulary and the ImageNet/Places categories.

Semantic Similarity Measure: The key of building robust semantic filters is the selection of filter words that have high semantic correspondence with the querying category. In this work, the word-wise semantic similarity between the reduced Google News vocabulary and the category is measured from the perspective of hyponymy and synonymy, respectively. Suppose that we are given a category query $y_q = \{s_q, \mathbf{w}_q\}$ from the ImageNet1000 basic categories and a candidate word $y_c = \{s_c, \mathbf{w}_c\}$ from the reduced Google News vocabulary, where s denotes the synset of y in WordNet taxonomy and $\mathbf{w} \in \mathbb{R}^v$ is the v -dimensional vector representation of y pre-trained by word2vec model. For the synsets s_q and s_c in Wordnet taxonomy,

¹ImageNet1000 dataset [48] contains 1.2 million object-centric images with 1000 categories (200 basic categories), and Places365-Standard dataset [43] as a subset of Places2 contains 1.8 million scene-centric images with 365 basic categories). Note that ImageNet1000 and Places2 employ both the fine-grained categories (e.g., “husky dog” or “German shepherd”) and the basic category (e.g., “dog”), following the WordNet hierarchy.

²are phrases longer than one word, e.g., “ice cream” is internally merged as a unigram expression “ice_cream”

TABLE 5.1: Semantic filters for object recognition and scene recognition. Four exemplars per category are shown in WordNet format.

Data Cluster	Category Sets	Semantic Filter Words
ImageNet	dog	puppy, prairie dog, guide dog, sheep dog
	antelope	impala, elk, wildbeest, gazelle
	laptop	notebook, laptop computers, notebook computers, computers
	pepper shaker	saltshaker, cruets, jar, castor
Places	subway station	subway, train station, railway station, underpass
	viaduct	overpass, pedestrian bridge, expressway, grade separation
	bank vault	vault, bank, depository, strongroom, deposit
	inn	cottage, guesthouse, farmhouse, manor house

Wu-Palmer (WP) similarity [122] is one of the most commonly used measures in computing the path-based concept relatedness of WordNet, which is based on the taxonomic distance with respect to their least common subsumer (LCS). The WP similarity score of these two synsets is derived by:

$$WP(s_q, s_c) = \frac{2 \cdot d(LCS)}{d(s_q) + d(s_c)}, \quad (5.3)$$

where $d(\cdot)$ denotes the taxonomic distance between the root of taxonomy and a certain synset, and the resulting $WP(\cdot, \cdot)$ is valued within the range from 0 to 1. For the $\mathbf{w}_q$ and $\mathbf{w}_c$ word vectors, on the other hand, we take the semantic relationship of synonymy into account and compute the cosine similarity between $\mathbf{w}_q$ and $\mathbf{w}_c$:

$$\cos(\mathbf{w}_q, \mathbf{w}_c) = \frac{\mathbf{w}_q \cdot \mathbf{w}_c}{\|\mathbf{w}_q\| \cdot \|\mathbf{w}_c\|}. \quad (5.4)$$

The cosine similarity computation acts as the bounded dot product of $\mathbf{w}_q$ and $\mathbf{w}_c$ to decrease variance. According to [136] and [4], the cosine similarity measure in vector space can be interpreted as the degree of one word semantically matching another word. To incorporate the cosine similarity and WP similarity measures, we restrain the cosine similarity score to vary within the same range as the WP similarity score, and therefore have:

$$\cos_{abs}(\mathbf{w}_q, \mathbf{w}_c) = \max[0, \cos(\mathbf{w}_q, \mathbf{w}_c)]. \quad (5.5)$$

For both similarity scores above, the higher value means the closer the word y_q towards the category query semantically, and vice versa. In this work, we consider the hyponymy similarity and the synonymy similarity are equally significant in measuring the relationship of y_q and y_c . Therefore, we assess the overall semantic similarity by taking the average of the cosine similarity and WP similarity scores:

$$A(y_q, y_c) = \text{mean} [WP(s_q, s_c), \text{cos}_{abs}(\mathbf{w}_q, \mathbf{w}_c)] . \quad (5.6)$$

Instead of using max or min operation, we average the WP and cos_{abs} scores in Eq. (5.6) because we find that the two aspects of lexical hyponymy and synonymy are actually complementary. For instance, the WP and cosine similarity scores of two semantically dependent words “husky” and “dog” are 0.87 and 0.43, respectively. Obviously, the word vectors based on contextual information fails to learn the strong semantic synonymy, whereas the lexical hyponymy of synsets represented by the WP similarity measure can help to capture the true semantic similarity of the two words with high similarity score. Conversely, the WP and cosine similarity scores of another two semantically independent words “husky” and “kitty” are 0.74 and 0.21, respectively. The lexical synonymy of word vectors represented by the cosine similarity measure can help to preserve the semantic diversity of the two words with low similarity score. Hence, an average operator on the WP and cos_{abs} scores is more suitable to achieve a tradeoff of lexical hyponymy and synonymy for semantic similarity evaluation.

By setting a hard threshold T (e.g., $T = 0.5$) on $A(y_q, y_c)$ score, we obtain h groups of preliminary filters corresponding to the category queries of the ImageNet1000 ($h = 200$ in this case). Some examples of the semantic filter abstraction are shown in Table 5.1. The first (upper) part of Table 5.1 refers to the semantic filter words obtained for object recognition problem, while the filter words for scene recognition problem are shown in the second (lower) part of the table. Finally, we flatten the filters from different groups and remove the replicated elements. As a result, the task-generic semantic filter bank $Y_B \in \mathbb{R}^z$ is established for object recognition problems, where z is the size of filter bank. Similarly, we build another scene-centric filter bank following the same procedure, with respect to the Places365 category queries. The procedure of semantic filters generation is illustrated in Algorithm 3.

By the analogy of Equation (5.2) and the semantic filters acquired above, the semantic filter then slides over every row vector of $\mathbf{X}$ unigram by unigram, and yields a bunch of filter responses $\{c_1, c_2, \dots, c_n\}$. Next, the filter responses are activated through non-linear functions like rectified linear units (“ReLU”, denoted as $\sigma(\cdot) = \max(0, \cdot)$). As a result, we obtain the feature map $\mathbf{f}$ for every semantic filter, which is denoted as:

$$\mathbf{f} = [\sigma(c_1), \sigma(c_2), \dots, \sigma(c_n)]. \quad (5.7)$$

Then $\mathbf{f}$ is fed into the pooling operation to generate the most salient text features for the s-CNN network. In the next section, we elaborate the novel double pooling scheme in s-CNN.

5.2.2 s-CNN with Double Pooling

The s-CNN model with the complete filter bank is structurally very similar to the CNN single-channel baseline proposed in [2], while the s-CNN Clustered models are more complex with a multi-channel architecture of convolution operations. Therefore, the pooling methods used for s-CNN and s-CNNC models are also different. Specifically, s-CNN model adopts one-step max-over-time pooling method, while the s-CNNC models involve max-over-time and max/average-over-cluster double pooling method. In the following, we firstly demonstrate the semantic filter clustering approach.

Semantic Filter Clustering: The two semantic filter banks mentioned above provide universal feature descriptors for object and scene recognition problems, respectively. However, our method inevitably uses more semantic filters than the conventional CNN to achieve good coverage of generic filters for various tasks. Hence, the semantic correlation within the semantic filter banks may increase due to the semantic overlaps of the semantic filters themselves. Intuitively, high semantic correlation of feature descriptors leads to feature redundancy. To boost classification generalization, it is necessary to reduce the dimensionality and redundancy. In this work, we propose to cluster the semantic filters within the filter banks prior to convolution. The dimensionality and redundancy concerns can be

Algorithm 3 Task-generic Semantic Filters Generation

Input: the category queries: $[y_q^{(1)}, \dots, y_q^{(h)}]$, where $y_q = \{s_q, \mathbf{w}_q\}$; the word pool from the reduced Google News vocabulary: $[y_c^{(1)}, \dots, y_c^{(l)}]$, where $y_c = \{s_c, \mathbf{w}_c\}$; the overall similarity threshold T ; required number of word clusters: m .

Output: a filter bank Y_B with size of z ; or filter clusters: $[Y_C^{(1)}, \dots, Y_C^{(m)}]$.

1. **for** $1 \leq i \leq h$ **do**
 - 1: **while** j **not exceed** l **do**
 - 2: Measure the Wu-Palmer similarity $WP(s_q^{(i)}, s_c^{(j)})$ on the synsets using Eq. (5.3).
 - 3: Compute the cosine similarity $\cos_{abs}(\mathbf{w}_q^{(i)}, \mathbf{w}_c^{(j)})$ on the word vectors using Eqs. (5.4) and (5.5).
 - 4: Obtain $A(y_q^{(i)}, y_c^{(j)})$ by averaging the WP similarity and cosine similarity scores as shown in Eq. (5.6).
 - 5: **end while**
 - 6: **if** $A(y_q^{(i)}, y_c^{(j)}) > T$ **then**
 - 7: Select $y_c^{(j)}$ as a semantic filter.
 - 8: **end if**
2. **end for**
3. Flatten the returned h groups of preliminary filters and remove the replicated elements to obtain the filter bank $Y_B = [y_b^{(1)}, \dots, y_b^{(z)}]$.
4. **if** filter clustering is performed **then**
 - 1: Measure the overall similarity within the semantic filter bank using Eq. (5.6) and obtain a similarity matrix $\mathbf{W}$.
 - 2: Apply K-means clustering method based on the overall similarity matrix by setting the number of clusters to m .
 - 3: **return** $[Y_C^{(1)}, \dots, Y_C^{(m)}]$
5. **else**
 - 1: **return** Y_B
6. **end if**

effectively addressed when the filters are used by clusters in the s-CNN network from the perspective of semantic similarity.

Supposed that we are given a semantic filter bank $Y_B \in \mathbb{R}^z$, we measure the overall pairwise semantic similarity between different semantic filters to obtain a semantic similarity matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{bmatrix} A(y_b^{(1)}, y_b^{(1)}) & A(y_b^{(1)}, y_b^{(2)}) & \dots & A(y_b^{(1)}, y_b^{(z)}) \\ A(y_b^{(2)}, y_b^{(1)}) & A(y_b^{(2)}, y_b^{(2)}) & \dots & A(y_b^{(2)}, y_b^{(z)}) \\ \vdots & \vdots & \ddots & \vdots \\ A(y_b^{(z)}, y_b^{(1)}) & A(y_b^{(z)}, y_b^{(2)}) & \dots & A(y_b^{(z)}, y_b^{(z)}) \end{bmatrix} \quad (5.8)$$

where $A(y_b^{(i)}, y_b^{(j)})$ is given by Eq. (5.6). Based on the overall similarity matrix, K-means clustering method is applied to find m filter clusters $[Y_C^{(1)}, \dots, Y_C^{(m)}]$, where $Y_C^{(i)} \in \mathbb{R}^{t_i}$ and $\sum_{i=1}^m t_i = z$. The number of clusters m can be set by users, while we make it equal to the number of category queries of the Image1000 and Places365 datasets for the object-centric filter clusters and scene-centric filter clusters, respectively. This default setting is validated experimentally in Section 5.4. The procedure of semantic filter clustering is also illustrated in Algorithm 3.

k -Max-Over-Time Pooling: To the s-CNN model with complete filter bank $Y_B \in \mathbb{R}^z$, we obtain a matrix of feature maps $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_z] \in \mathbb{R}^{z \times n}$ from the single-channel convolutional layer. We then apply one layer of k -max pooling over the feature maps to obtain a feature matrix:

$$\mathbf{U} = [\mathbf{p}_{\max}(\mathbf{f}_1), \mathbf{p}_{\max}(\mathbf{f}_2), \dots, \mathbf{p}_{\max}(\mathbf{f}_z)] \in \mathbb{R}^{z \times k} \quad (5.9)$$

where

$$\mathbf{p}_{\max}(\mathbf{f}) = \begin{bmatrix} \max_1(\mathbf{f}) \\ \vdots \\ \max_k(\mathbf{f}) \end{bmatrix}, \quad (5.10)$$

$\max_i(\cdot)$ extracts the i -th highest values of $\cdot$. Compared to simple 1-max pooling [137], the k -max pooling enables us to extract the k most discriminative feature rather than pool out just one active feature. In addition, the k -max pooling scheme neglects the filter responses with less semantic relatedness, playing an core role in the noise suppression along with the input word vector mapping process.

When similar semantic filters are clustered, we deploy a multi-channel architecture with m channels of convolution. From each channel, we obtain the feature maps $\hat{\mathbf{F}} = [\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \dots, \hat{\mathbf{f}}_t] \in \mathbb{R}^{t \times n}$, where t denotes the number of semantic filters applied. Next, we apply the same k -max pooling over the feature maps $\hat{\mathbf{F}}$ to obtain the feature matrix $\hat{\mathbf{U}} \in \mathbb{R}^{t \times k}$ for a channel with cluster of filters.

When similar semantic filters are clustered, we deploy a multi-channel architecture with m channels of convolution. From each channel, we obtain the feature maps $\hat{\mathbf{F}} = [\hat{\mathbf{f}}_1, \hat{\mathbf{f}}_2, \dots, \hat{\mathbf{f}}_t] \in \mathbb{R}^{t \times n}$, where t denotes the number of semantic filters applied. Next, we apply the same k -max pooling over the feature maps $\hat{\mathbf{F}}$ to obtain the feature matrix $\hat{\mathbf{U}} \in \mathbb{R}^{t \times k}$ for a channel with cluster of filters.

Max/Average-Over-Cluster Pooling: The s-CNN model with complete filter bank employ max-over-time pooling only. In the context of filter clustering, we add a successive max/average-over-cluster layer to the feature matrix $\hat{\mathbf{U}}$ to reduce the representation dimensionality. In the max/average-over-cluster layer, we propose two methods to merge the features and build two variants of s-CNN Clustered models named as s-CNNC_{max} and s-CNNC_{ave} by applying max and mean operations to $\hat{\mathbf{U}}$ column by column over the cluster of filters:

$$M(\hat{\mathbf{U}}) = \begin{cases} \max(\hat{\mathbf{U}}) & \text{for s-CNNC}_{\max} \\ \text{mean}(\hat{\mathbf{U}}) & \text{for s-CNNC}_{\text{ave}} \end{cases} \quad (5.11)$$

The use of filter clusters instead of individual semantic filters is driven by the idea of reducing feature redundancy. In particular, we apply an additional feature merging layer in the context of filter clustering. While both s-CNNC_{max} and s-CNNC_{ave} models effectively reduce the feature dimensionality and redundancy, their mechanisms are different. In s-CNNC_{max}, max operation reduces intra-class spread and identifies the features that most relevant to every basic category set, whereas mean operation in s-CNNC_{ave} model neglects the intra-class spread and takes the centroids of all features with respect to every basic category set. We compare the performance of s-CNNC_{max} and s-CNNC_{ave} for classification experimentally in Section 5.4.

The features obtained from the pooling layer(s) are then passed to a fully connected softmax classifier with dropout regularization to avoid over-fitting. The semantic filters are represented by the pre-trained vectors and not trained in the network, in a similar way of the transferable CNN models for pre-trained image representation. We also compare the performance of the s-CNN model without additional pooling method and the s-CNNC models in the experiments. The results are shown in Table 5.2.

5.3 Multimodal CNN

5.3.1 Multimodal CNN for Joint Representation Learning

The proposed multimodal CNN framework consists of image CNN and text-based s-CNN that interact mutually to learn the text-image joint representation by an end-to-end training method. From a macro perspective, this framework is also a prototypical example of feature-level fusion strategy. The text representation is learned via the s-CNN model as demonstrated in the previous section, while the image representation is transferred from the pre-trained VGG-very deep 16 [48] model. To handle this cross-domain scenario, fine-tuning approach is utilized to pursue more robust multimodal model. As shown in the lower part of Figure 5.1, we take the pre-trained VGG16 model as an example and remove the output layer FC16 of the network. Then, we add an adaptation layer formed by a fully-connected layer named FCa after the FC15 layer. Unlike the conventional fine-tuning methods that consider only image representation, the FCa adaptation layer here obtains the presence of text modality data by mutual learning with semantic convolutional activations under the aggregated framework. By minimizing the multi-class logistic regression loss (a.k.a cross-entropy loss) with respect to the output of softmax layer, a backward propagation is applied to update all the parameters iteratively in the unified multimodal CNN framework. During this multimodal learning process, the interaction between the image CNN and text s-CNN is successfully encoded to the text-image joint representation.

5.3.2 Multimodal CNN with Noise Reduction

Considering external web text data is used to aid image classification, it will inevitably bring in noise. In general, the noise here is the irrelevant text with regard to the task-of-interest, which can easily nullify the benefits achieved by the multimodal CNN (mm-CNN) framework. To alleviate the negative impact caused by the noisy text, it is necessary to assess whether the auxiliary text data should be used and learned prior to the mm-CNN network training.

To this end, we adopt the semantic matching with z task-generic semantic filters (mentioned in Section 5.2.1) to obtain the semantic matching scores of the external

text data. The semantic matching scores are computed by the cosine similarity score between the input text and semantic filters using Equation (5.4). Besides the semantic matching score, the occurrence of words that corresponding to each semantic matching score is considered to be equally important according to the insights of the BoW models. Suppose that we are given an arbitrary text data with the semantic matching score vector and its corresponding word occurrence vector, the occurrence-based semantic matching scores are computed by taking the element-wise average of the two vectors $\mathbf{o}$ and $\mathbf{p}$, which can be represented by $A(\mathbf{o}, \mathbf{p})$ ($A(\cdot, \cdot)$ is given by Eq. (5.6)). The vectors $\mathbf{o}$ and $\mathbf{p}$ are denoted as:

$$\mathbf{o} = \left[\frac{o_1}{\max(\mathbf{o})}, \frac{o_2}{\max(\mathbf{o})}, \dots, \frac{o_z}{\max(\mathbf{o})} \right] \quad (5.12)$$

and

$$\mathbf{p} = [p_1, p_2, \dots, p_z] \quad (5.13)$$

where o_i denotes the occurrence of the word that with highest semantic matching score with respect to i -th semantic matching score p_i , which is determined by the term frequency in the text. The value of o_i is scaled by $\max(\mathbf{o})$ to restrain the word occurrence to vary within the same range as semantic matching scores. Then, we compare the maximum value of $A(\mathbf{o}, \mathbf{p})$ with a threshold T (e.g., $T = 0.5$). As a result, the external text data is selected for the multimodal CNN training only if $\max A(\mathbf{o}, \mathbf{p}) > T$.

In practice, we use the selected subset of the external text data and its corresponding subset of image data only to train the multimodal CNN model as shown in Figure 5.1. Eventually, the results obtained from the mm-CNN model on the selected subset of image/text data and the results obtained from the visual CNN model on the unselected subset of image data are combined on decision level during testing. We experimentally evaluate the significance of applying noise reduction scheme for mm-CNN training and find it can produce further improvements compared to the mm-CNN training without noise reduction.

5.4 Experiments

In this section, we validate the effectiveness of the proposed s-CNN and s-CNNC models for text representation learning, as well as the multimodal CNN framework

on the scene recognition and object recognition problems symmetrically.

5.4.1 Datasets

We firstly describe the six datasets used in the experiments, including three scene-centric benchmarks : Scene-15 [20]; MIT Indoor-67 [138]; UIUC-Sports [111], and three object-centric benchmarks: Caltech-256 [112]; Pascal Visual Object Classification (VOC)-2007 and VOC-2012 [139]. Note that four out of six datasets were applied for experimental implementations in Chapter 3 and Chapter 4. Hence, we briefly introduce the two newly added MIT Indoor-67 and VOC-2007 datasets in this section³.

- **MIT-67** consists of 67 categories of indoor scene images. The same as experimental settings in [138], the experiments are run based on a subset of the dataset (80 images per category for training and 20 images for testing as given in [138]). The classification on MIT Indoor-67 is very intractable due to the large inter-class variation.
- **Pascal VOC** challenges have annually updated the 20 object-category repository since 2007 to 2012. Here, we evaluate the classification performance on VOC-2007 and VOC-2012. The VOC-2007 dataset has 9,963 images, while the VOC-2012 dataset employs more than 22,500 images. The VOC datasets are roughly split into 50% for training/validation and 50% for test. The Pascal VOC challenges have up to 6 objects of different categories for each image, where multi-label image classification is a more practical scenario in the real-world problems, and therefore deserves well exploration. However, it usually encounters the absences of multi-object alignments and interactions as different objects in certain image may have variant scales, spatial occlusion, and etc.

The external web text data used for the above six image datasets is obtained by online web retrieval method proposed in Chapter 3 and the data preparation process is the same as the approach elaborated in Chapter 4.

³The details and statistics of the other four datasets have been illustrated in Section 3.4 of Chapter 3

5.4.2 Experimental Setup and Evaluation Metrics

Our proposed s-CNN and its clustered variants are compared with the following text representation methods:

- BoW: the TF-IDF representation (sparse 1-of- V encoding vectors in high dimensions) based on the clean web text are fed into a softmax classifier.
- CNN_{ns}: we re-implement the CNN-non-static model [2] on the web data. Specifically, filter windows of 1, 2 with 100 feature maps each are adopted to learn the semantic dependency based on unigram words and bigram phrases.
- Word embeddings: we utilize the word2vec vectors pre-trained on the Google News corpus to represent the web text and the semantic filters. The pre-trained word vectors used in this work have a dimensionality of 300, which are fine-tuned with respect to specific tasks by the proposed s-CNN models.
- s-CNN_{max}: we also verify the importance of feature redundancy reduction by testing on the basic s-CNN model without filter clustering and additional feature merging layer.
- s-CNNC_{max} and s-CNNC_{mean}: our proposed s-CNN models with semantic filter clustering and additional feature merging (max and mean merging methods).

Semantic Filter Generation: We set the threshold value on the overall semantic similarity $A(y_q, y_c)$ of Equation (5.6) as $T = 0.5$, which is the same threshold value for noise reduction scheme in Section 5.3.2. As a result, 1,500 and 1,817 generic filter words are finally selected for text-visual object recognition and scene recognition, respectively. In terms of the required numbers of object-centric and scene-centric filter clusters, we search the appropriate m values for object and scene recognition around the number of basic categories Q of ImageNet1000 and Places365 (i.e., $\{Q - 100, Q, Q + 100\}$), respectively. Accordingly, we conducted three individual experiments on two datasets using s-CNNC_{max} for the above two types of problems respectively. The contrastive results shown in Figure 5.2 illustrate that $m = 200$ and $m = 365$ are the best overall settings of semantic filter clustering for object recognition and scene recognition, respectively.

FIGURE 5.2: Contrastive s-CNN_{max} experiments to find m within $\{Q - 100, Q, Q + 100\}$. (a): object recognition; (b): scene recognition.

Model Training: hyperparameters of s-CNN for text representation are defined as follows. Semantic filter window size is constantly 1. We employ rectified linear units (ReLU) as non-linear activation function, RMSprop as the optimizer for mini-batch gradient decent with learning rate of 0.001, the mini-batch size of 50, dropout rate of 0.5, and the value k in k -max pooling as 5. The FCa layer of multimodal CNN model is with a dimensionality of 2,048.

Pre-trained Image CNN Models: three publicly accessible CNN models are used in this work. Intuitively we perform softmax classifiers on the off-the-shelf CNN visual features and show their results as “VGG”⁴ [129], “ResNet152” model [50], and “Xception” model [125]. Their multimodal variants interleaved with the s-CNN_{max} are denoted as “s-CNN+VGG”, “s-CNN+ResNet”, “s-CNN+Xception” below, respectively. Note that we use the publicly available CNN models to extract deep visual features on the whole images. Since learning image representation is not the main focus of our work, none of the prevalent data augmentation techniques is involved in our implementation.

Evaluation Metrics: For Scene-15, Caltech-256, and UIUC-Sports datasets, the experiments are run on 10 random training/test splits and the mean accuracy is

⁴VGG16 model trained on the Hybrid-1365 dataset [43] (a combination of ImageNet ILSVRC1000 and Places365-Standard)

TABLE 5.2: Comparison of top-1 accuracy (%) to the comparing approaches on the text representation of 4 different datasets. The first three datasets are scene-centric benchmarks, while the last one is an object-centric dataset. The starred methods are the methods shown in Chapter 4.

Method	Scene-15	MIT-67	UIUC-8	Cal-256
BoW*	70.1	52.5	83.5	63.4
CNN _{ns} [2]	72.2	52.0	85.9	65.2
SMNN*	75.3	-	90.4	66.5
s-CNN _{max}	76.2	54.6	90.9	67.3
s-CNNC _{ave}	76.7	55.1	91.2	66.6
s-CNNC _{max}	77.2	55.9	91.5	68.3

evaluated. In this work, we also evaluate and validate the statistical significance of the classification performance following the same paired t -tests as shown in Chapter 3 and Chapter 4. In addition, the top-1 accuracy is reported on the MIT-67 test set. According to the Pascal challenge protocols, the performance on the VOC-2007 and VOC-2012 is evaluated by average precision (AP). Since the ground-truth labels of VOC-2012 test data are not publicly available, we train the classification models on the training/validation set and upload the experimental results of the test set to the official server for evaluation.

5.4.3 Results on Semantic Convolutional Neural Network

We show the experimental results of different text modeling methods on the six benchmark datasets in Table 5.2 and Table 5.3, where the external web text data retrieved from images is tested in the experiments. The results shown are obtained from the softmax classifier posteriors. In terms of the metrics, we firstly compare the top-1 accuracy (mAcc) scores of our proposed s-CNN method on the first four datasets and two baseline text models. And then the classification precision (mAP) scores on the two VOC datasets are evaluated and compared.

In the upper part of Table 5.2 and Table 5.3, it is observed that s-CNN_{max} model without feature merging layer outperforms the other text modeling methods on multiple benchmarks, suggesting that the semantic filters proposed can universally extract features for various classification tasks. It also proves that the semantics

TABLE 5.3: Comparison of mAP (%) to the comparing approaches on the text representation of VOC-2007 and VOC-2012 object-centric datasets.

Method	VOC-2007	VOC-2012
BoW*	65.5	66.8
CNN _{ns} [2]	71.1	72.1
SMNN*	-	71.8
s-CNN _{max}	72.8	72.4
s-CNNC _{ave}	73.1	73.2
s-CNNC _{max}	73.5	74.1

encoded in the filters are actually helpful to boost CNN representation and classification. Compared to the s-CNN_{max} results, the s-CNNC_{max} and s-CNNC_{mean} models usually achieve further improvements in the contrastive experiments as shown in the lower part of Table 5.2 and Table 5.3. The better performance achieved by the s-CNNC models is attributed to the better s-CNN representation with less feature redundancy and lower dimensionality. Furthermore, the performance of the s-CNNC_{max} and s-CNNC_{mean} models is also compared. We observe that s-CNN_{max} models generally surpass s-CNN_{ave} models on all datasets by a slim margin, suggesting the importance of max merging over mean merging for reducing intra-class spread. Next, we will show that the s-CNN models combined with image CNN can further facilitate the image recognition.

5.4.4 Results on Scene Recognition

Table 5.4 shows the classification accuracy scores on the three scene-centric datasets for various approaches. The first (upper) part of Table 5.4 shows the performances of the methods that use hand-crafted visual features, such as (improved) Fisher Vector [27] shallow features. And the second (lower) part of the table shows the classification accuracy of popular deep features of the pre-trained CNN models. Compared with the results obtained by the hand-crafted features including IFV features (the champion in the ILSVRC2012 competition), the deep features from the CNN models achieve impressive results, overwhelming the shallow features in all the three datasets. The success of these CNN models is attributed to training on large scale auxiliary image repositories.

Furthermore, we compare the proposed multimodal CNN approach with the multi-feature learning methods on the three scene-centric datasets. The methods listed

TABLE 5.4: Top-1 classification accuracy (%) on the Scene-15, MIT Indoor-67 and UIUC-Sports datasets for the state-of-the-art methods and our methods.

The starred result is the best performance shown in Chapter 4.

Method	Scene-15	MIT-67	UIUC-Sports
Object Bank [108]	80.9	37.6	76.3
LScSPM [24]	89.7	-	85.3
DDSFL [127]	86.9	52.3	84.4
IFV [27]	89.2	60.8	90.31
ISPR [131]	85.1	50.1	74.9
ISPR+IFV [131]	91.1	68.5	92.1
CNN-sNBNL [140]	92.9	72.8	95.3
VGG [43]	93.0	75.3	95.6
ResNet152 [50]	92.4	77.4	96.9
Xception [125]	92.5	76.2	97.2
CNN fusion [132]	90.7	70.1	94.7
DDSFL+Caffe [127]	92.8	76.2	96.8
ResNet+Layout [141]	94.5	81.0	-
SMNN+Xception*	96.5	-	98.9
s-CNN+VGG	94.8	78.5	97.1
s-CNN+ResNet	96.4	81.6	98.2
s-CNN+Xception	97.2	80.4	98.7

in the last column of Table 5.4 involve multi-feature fusion as opposed to the single-feature learning methods that were discussed above. Unlike the first three fusion methods that merge different visual features, the SMNN+CNN method and our proposed multimodal CNN models learn the feature representations of different modality. Combined with the s-CNN_{C_{max}} methods, the multimodal CNN models with whole set of external text data achieves excellent classification results across all three scene-centric experiments, in which “s-CNN+VGG”, “s-CNN+ResNet”, and “s-CNN+Xception” outperform their purely visual CNN methods by 2.5%, 2.8%, and 2.8% in overall accuracy averaged over the three datasets, respectively.

5.4.5 Results on Object Recognition

Besides scene recognition problems, we also evaluate the effectiveness of our proposed approach for object recognition problems. A great deal of images in the object-centric Pascal VOC datasets contain more than one category label, which corresponds to different objects in a single image. Therefore, we divide the object

TABLE 5.5: Top-1 Classification accuracy (%) on the Caltech-256 dataset. Note that the method listed with $\hat{}$ denotes that only a “subset” of web text data is included, which is selected by the mm-CNN with noise reduction.

Method	# training data	
	15	60
ScSPM [109]	27.7	40.1
LLC [23]	32.7	46.1
Deep SIFT+LLC [130]	34.6	42.8
5-layer ConvNet [128]	65.0	74.2
DeConvNet [129]	67.0	77.9
VGG [43]	70.9	76.9
ResNet152 [50]	68.7	78.2
Xception [125]	73.4	80.6
s-CNN+VGG	73.8	84.1
s-CNN+ResNet	73.2	83.0
s-CNN+Xception	74.8	84.9
SMNN+Xception*	75.3	84.7
s-CNN+Xception $\hat{}$	75.6	85.4

recognition experiments into traditional single-label classification and multi-label classification on Caltech-256 dataset and VOC datasets, respectively.

Single-Label Classification: For the Caltech-256 dataset, we show the performance of different methods in Table 5.5. The first part of Table 5.5 lists the top-1 accuracy of the stat-of-the-art methods based on shallow image representations, while the remaining methods in this table involves the deep CNN visual features. Similar to the experimental results shown for scene recognition, the deep CNN features outperform the shallow features by a large margin regardless of discrepant amount of training data. With the number of training data increases, sufficient training samples result in further gain in accuracy for different image representations. Notably, an additional “s-CNN+Xception $\hat{}$ ” experiment with selective text data involved is performed and achieve new benchmark results on the Caltech-256 object recognition task, indicating that the noise reduction scheme proposed in multimodal CNN network is helpful. However, noise reduction method can not make a difference in the scene-centric experiments. The results with or without noise reduction are basically the same as shown in Table 5.4 because only less than 1% of the whole set was stripped out as noise. Therefore, the results of “s-CNN+Xception $\hat{}$ ” for scene-centric problem are not shown.

Multi-Label Classification: Different from the previous experiments discussed above, here we test on the challenging but practical VOC multi-label classification tasks. Table 5.6 and 5.7 demonstrate the per-class average precision (AP) and the overall mean AP (mAP) of the listed approaches on the two VOC datasets. The first (upper) part of them shows the results of methods that only image modality data is adopted, while the second (lower) part of the tables shows the results of methods that introducing fusion of text modality data.

TABLE 5.6: Per-class average precision (%) comparison for the listed methods on VOC-2007 test dataset. Note that the method listed with ^ denotes that only a “subset” of web text data is used.

Method	plane	bike	bird	boat	btl	bus	car	cat	chair	cow
FV [27]	75.7	64.8	52.8	70.6	30.0	64.1	77.5	55.5	55.6	41.8
CNN-SVM [142]	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5
HCP1000 [143]	95.1	90.1	92.8	89.9	51.5	80.0	91.7	91.6	57.7	77.8
VGG [48]	97.6	91.3	95.2	93.3	57.9	86.3	90.7	93.4	59.6	78.5
ResNet152 [50]	97.9	91.9	95.4	92.9	59.8	87.7	91.6	95.5	64.1	85.8
Xception [125]	98.5	87.4	96.6	94.8	62.7	85.5	92.8	98.3	71.3	94.4
HMF [3]	97.4	90.2	93.9	91.5	53.3	83.6	90.6	91.8	54.3	78.8
s-CNN+VGG	98.1	91.8	95.5	94.6	57.9	88.1	91.9	95.2	61.9	86.1
s-CNN+ResNet	98.7	91.9	95.4	94.4	60.6	88.7	92.4	96.0	64.3	88.1
s-CNN+Xception	98.8	94.1	96.6	96.9	64.0	92.3	92.0	97.4	66.2	93.7
s-CNN+Xception^	99.0	94.7	96.5	97.0	66.7	92.5	91.6	97.4	66.7	94.3

Method	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	mAP
FV [27]	56.3	41.7	76.3	64.4	82.7	28.3	39.7	56.6	79.7	51.5	58.3
CNN-SVM [142]	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.4	71.8	73.9
HCP1000 [143]	70.9	89.3	89.3	85.2	93.0	64.0	85.7	62.7	94.4	78.3	81.5
VGG [48]	79.2	93.5	93.1	89.2	94.7	60.1	87.5	67.7	96.6	78.1	84.2
ResNet152 [50]	79.9	95.1	95.5	90.6	95.6	63.5	86.8	74.9	97.3	79.0	86.0
Xception [125]	76.4	94.1	96.2	93.1	92.5	61.0	92.4	78.2	98.5	78.7	87.1
HMF [3]	77.5	92.8	94.3	88.7	90.4	56.7	83.4	63.6	96.4	70.9	81.9
s-CNN+VGG	80.7	95.1	95.8	89.9	94.9	60.4	90.1	70.9	97.2	79.1	85.4
s-CNN+ResNet	82.7	95.8	96.1	91.9	95.1	63.5	90.5	77.4	97.6	79.6	86.9
s-CNN+Xception	84.1	96.4	97.1	94.2	93.0	61.1	95.7	78.5	98.6	83.5	88.6
s-CNN+Xception^	83.7	96.3	97.4	94.5	93.3	60.6	95.9	78.3	98.8	84.3	89.0

In table 5.6, the performance of hand-crafted FV [27] features lags behind the CNN-based image representation by a large margin on the VOC-2007 dataset. Combined with the s-CNN_{C_{max}} text modeling method, the three multimodal convolutional models with whole set of external text data improve the mAP results by as much as 1.5%. Furthermore, “s-CNN+Xception^” trained on concise subset of external text data achieves the best AP scores over in 11 categories of the VOC-2007 dataset.

Table 5.7 shows the results of our multimodal convolutional network on the more challenging VOC-2012 dataset, which is illustrated in the same way of Table 5.6. Although the DDSFL features aim at exploring the class correlation within the image content, the effectiveness of hand-crafted pixel-based visual features is still not comparable to that of CNN-based mid-level features. In addition to the single modality DDSFL features, the direct concatenation of DDSFL and Caffe (i.e., a CNN model) features leads to a satisfactory progress in performance. More significantly, “s-CNN+Xception[^]” consistently outperforms the image-only CNN models over 8 categories of the VOC-2012 dataset.

TABLE 5.7: Per-class average precision (%) comparison for the listed methods on VOC-2012 test dataset. The starred method denotes the methods shown in Chapter 4.

Method	plane	bike	bird	boat	btl	bus	car	cat	chair	cow
DDSFL [127]	77.7	42.5	45.4	53.3	24.0	72.2	50.6	54.2	47.4	26.0
DDSFL+Caffe	92.7	75.4	85.1	79.4	42.3	88.1	68.5	87.1	62.9	65.8
HCP1000 [143]	97.7	83.0	93.2	87.2	59.6	88.2	81.9	94.7	66.9	81.6
VGG [48]	98.4	87.5	95.3	91.2	62.5	92.9	78.7	96.7	70.9	82.8
ResNet152 [50]	98.3	87.5	96.5	91.0	67.2	92.8	79.1	97.5	72.8	87.3
Xception [125]	99.1	88.4	96.3	91.3	69.3	94.1	79.9	98.5	72.9	92.5
HMF [3]	97.4	86.2	93.2	88.6	56.9	92.6	78.1	95.3	65.0	83.1
SMNN+Xception*	99.3	91.6	97.3	93.6	72.7	95.2	82.0	99.0	75.4	96.1
s-CNN+VGG	98.7	89.6	95.7	91.9	66.6	93.1	80.6	97.1	72.5	89.6
s-CNN+ResNet	99.0	89.3	96.5	91.5	70.0	93.1	81.5	97.6	74.9	92.1
s-CNN+Xception	99.2	91.2	96.7	92.5	74.0	94.5	81.3	98.7	75.5	96.2
s-CNN+Xception [^]	99.3	91.7	97.3	92.8	74.2	94.9	83.1	98.9	75.6	96.0

Method	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	mAP
DDSFL [127]	36.6	42.6	38.9	50.7	70.3	23.4	36.2	36.4	61.6	49.0	46.9
DDSFL+Caffe	60.6	80.2	74.0	76.4	92.3	43.6	74.3	53.2	88.4	73.7	73.2
HCP1000 [143]	68.0	93.0	88.2	87.7	92.7	59.0	85.1	55.4	93.0	77.2	81.7
VGG [48]	66.2	95.4	90.8	88.9	94.7	54.8	88.3	63.8	96.1	83.6	84.0
ResNet152 [50]	69.4	95.8	91.9	89.7	95.6	58.5	91.4	64.8	96.3	84.9	85.4
Xception [125]	72.2	96.9	97.4	91.5	93.9	54.5	95.6	66.5	97.1	87.5	86.7
HMF [3]	59.1	92.8	91.2	88.0	91.1	53.9	87.2	57.5	94.4	79.3	81.6
SMNN+Xception*	75.4	98.1	98.0	93.0	94.4	57.5	97.3	70.9	97.8	88.5	88.7
s-CNN+VGG	67.2	96.1	94.5	89.6	94.9	57.9	92.2	64.9	96.2	84.3	85.6
s-CNN+ResNet	68.9	96.6	94.8	90.2	95.3	61.7	94.5	66.9	96.2	85.4	86.9
s-CNN+Xception	74.5	97.6	97.3	92.4	93.7	59.2	96.9	70.4	97.5	88.5	88.5
s-CNN+Xception [^]	82.5	97.7	97.9	94.1	93.8	59.5	96.8	71.1	97.6	88.4	89.1

5.5 Conclusions

In this chapter, we have proposed a novel text modeling prototype, called semantic convolutional neural network (s-CNN) for high-level text representation abstraction. The proposed s-CNN model implements universal semantic filters for various specific tasks under the same type of image classification problems. To reduce feature redundancy caused by semantic overlaps of semantic filters, the adoption of filter clustering methods are introduced in this chapter. The experimental results show the overwhelming performance of s-CNN over CNN on the text representation of text-visual recognition problem. And the s-CNN clustered variants s-CNNC models can achieve further improvement with less feature redundancy. Moreover, we incorporate the visual CNN models with the proposed textual s-CNN model under a feature-level fusion framework, which is able to be trained end-to-end. In addition, the use of noise reduction method can make the multimodal CNN models more robust to the noise effect. Experimental results also show the good complementarity of CNN-based image representation and s-CNN-based text representation, and the multimodal convolutional models could significantly boost scene and object classification in feature level.

Recently, Gan et al. [144, 145] employed web images and videos to train deep neural networks for video recognition. The proposed SMNN can be combined with these work to learn multimodal attention models and produce more salient and semantic representations for enhanced video recognition.

Chapter 6

Adaptive Attention Network for Web Text-Aided One-Shot Image Classification

Different from the previous chapters that investigated web text for aiding image classification in general, this chapter discusses an extreme case in image classification problem that is identifying the images when very little supervised data is provided. A typical example of this in the literature is the one-shot learning setting. In particular, we attempt to make use of the web text to aid this one-shot learning problem.

One-shot learning has recently drawn a great attention in the field of visual recognition, including image classification and retrieval. Unlike strongly supervised learning prototypes, one-shot learning has the extraordinary ability to recognize unfamiliar images from a single observation with labeling information. This is actually inspired by human cognition of rapidly learning new concepts with very little visual clues from memory fragments or external references. Mimicking humans' rapid learning process, we propose an novel model to learn visual attentions (clues) for one-shot learning via the text information obtained from external web resources. The web resources (e.g, text) has been largely omitted in the existing one-shot learning methods. This is presumably because the use of unstructured web data will inevitably bring in noise to their models. To address this concern, we develop an adaptive mechanism that is flexible to decide whether to attend to the visual

clue informed by the web text or to the image content itself. Under the one-shot settings, we show the practical importance of adaptive web text-aided attention model in classifying both the object-centric images and the more complex indoor scene-centric images.

6.1 Introduction

In this section, we firstly introduce the motivations of our work on one-shot learning. Then, the related work in one-shot learning and attention networks is reviewed briefly.

6.1.1 Motivations

The size of training data with human-labeled annotations is always one of the key issues in conventional supervised methods for visual recognition problems. Nowadays, deep learning has been performing on hundreds of thousands of training data to achieve good generalization. In practice, however, some extreme situations may have little labeled data available for training. In face recognition systems, for example, our practical goal is to recognize a persons identity by just feeding one picture of that persons face to the system. How to break through the bottleneck of generalization on very limited labeled data becomes a compelling challenge. A prototypical remedy to this problem is one-shot learning [146, 147], which is able to identify unlabeled data from novel classes with the aid of single labeled observation from each novel class (hereinafter called “exemplars”).

Actually, humans is able to recognize unknowns rapidly based on very few examples from the reference books or symbolic memory. Similarly, one-shot learning attempts to achieve effective and fast adaption to the unlabeled data with visual clues learned from little supervised information. Recently, Vinyals et al. [148] employ memory-augmented network that maps a set of exemplars and every new observation to its corresponding label, which greatly alleviates the data-hungry need for fine-tuning the deep neural network. Wang et al. [4] accommodate class tag information in the attention network training for one-shot classification. The advantage of text-based attention model in one-shot regime is that the attention

map generator learned from the training data (hereinafter called “auxiliary set”) can be generalized and transferred to the test data from novel classes (hereinafter called “evaluation set”). Nevertheless, using specific class tag may restrict the generalization of attention mechanism. For example, the semantic difference between class tag “inside subway” and “subway” in MIT-Indoor67 dataset is extremely vague, which may produce attended image features with less discriminative power.

In this work, we attempt to build an attention model that is not affected by the semantic correspondence of the class tags. Our approach is motivated by the web text-aided frameworks that learn semantic text features for image classification [3, 15]. Similar to these methods, we firstly explore the visually similar web images of target images, and then use their textual descriptions as web text data of the target images. In the training stage, the labels of the training images are directly transferred to the their corresponding unlabeled web text following the assumption presented in [6, 149]. In the attention encoder-decoder process, we learn the semantic coding vectors from the web text data via the task-generic semantic convolutional neural network (s-CNN), which has been introduced in Chapter 5. Therefore, the text-aided information used in this work is task-generic and not limited to the semantic correspondence of the class tags.

In many cases, the web text is notoriously noisy, which can easily mislead the visual-text attention model and nullify the overall performance. To address this problem, we develop an adaptive mechanism to automatically decide when to attend to the visual clues informed by the attention map generator or count on image features. In summary, the major novelties and contributions of this work are:

- We show that informative web resources can be employed to facilitate image classification in one-shot setting without knowing the class tags of the novel classes.
- We design an adaptive attention model to diminish the negative impact caused by the noisy components of the web text information, which makes our model more robust and effective.
- We validate the performance of our proposed attention network on both object and scene classification problems, and our methods outperforms the baseline methods significantly.

6.1.2 One-Shot Learning

In recent years, the research into one-shot learning has received considerable attention due to its weakly supervised characteristic. In this section, the related work in one-shot learning regime is briefly reviewed. The history of one-shot learning can be traced back to early 2000's with work by Li Fei-Fei et al. [146, 150]. Using a few images from the learned object classes, Fei-Fei et al. incorporated generic knowledge obtained from learned models of novel classes via a variational Bayesian framework. Specifically, the generic knowledge is used as the prior probability in a probabilistic model. Given one exemplar of the novel classes a posterior density is generated by updating the prior in the light of one observation. Some related work to the Bayesian framework for one-shot learning is [147, 151].

Another research direction upon weakly supervised problems addresses one-shot learning using distance-based metric learning, which maps the generalized knowledge learned from the labeled training data to the unseen or novel classes via learning the latent or embedding space. Thus, the distance between the exemplar and the evaluation set (both from novel classes) can be used to measure the similarity. Koch et al. train a Siamese neural network [152] that can identify pairwise inputs per novel classes and verify whether they belong to the same class via feature mapping. Santoro et al. [153, 154] develop memory-augmented neural networks that can rapidly transfer and assimilate the sufficient information in the short- and long-term memory to new tasks. Vinayals et al. [148] present a matching network method, which employs an attention mechanism over a learned embedding of the labeled training set to predict the evaluation set of unlabeled novel classes. Prototypical networks [155] learn a metric space in which classification is performed by assessing distances to per-class prototype representations.

6.1.3 Attention Mechanism

Unlike one-shot learning that is a relatively immature, attention mechanism has been rapidly developed and incorporated into deep neural networks for a wide range of applications in vision related domains, including image classification [156–158], image captioning [159, 160], image segmentation [161, 162], face modeling [163], visual tracking [164] and etc. Regardless the difference of application and focus,

they have similar objectives of using attention mechanism. These attention models can rapidly attend to and locate certain specific image regions with salient visual clues [165]. The most related attention network to the present work is that of Wang et al. [4], which takes class tag information into account for one-shot learning. Our proposed adaptive attention mechanism is inspired by the idea of visual sentinel in [160], and the details are presented next.

6.2 Proposed Method

In this section, we firstly describe the motivation and problem formulation of the proposed method presented in this work, then introduce the details of the web text-aided encoder-decoder attention model as well as the advanced adaptive attention model for one-shot image classification.

6.2.1 Problem Definition

One of the most important and attractive advantages of one-shot learning is that it can effectively relieve the shortage of labeled training data, which is usually obtained from labor-expensive and time-consuming human annotation. One-shot learning is motivated by the fact that humans can generalize unknown visual signals with weak supervision by linking with their symbolic memory [166] or seeking for help from external references [104], such as books and Internet. The latter pipeline motivates us to study on the external web text-aided approach to see whether it can contribute to one-shot learning in artificial intelligence. Our attention model leverages the relevancy of the visual features and text features to the learn attention-based mapping function. This attention mapping is generalized to the evaluation data of novel classes.

Assuming that we are given a labeled image instance $I = \{\mathcal{X}, \mathcal{X}^*, \mathbf{z}\}$ belonging to the class $h \in \mathcal{H}^a$, where $\mathcal{X}$ is a set of spatial visual features learned from the last convolutional layer of VGG16 network [48]: $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$, $\mathcal{X}^*$ is the global CNN feature vector obtained by taking global average pooling over $\mathcal{X}$, $\mathbf{z} \in \mathbb{R}^l$ is the text feature vector extracted from the web text of visually similar web images using s-CNN and $\mathcal{H}^a$ denotes the category set of the auxiliary set $\mathcal{A}$. The attention

FIGURE 6.1: Overview of the semantic-level feature extraction and encoding. The four examples of web visually similar images are obtained using online-based image search [3], and the text descriptions of them are aggregated and represented by pre-trained word vectors.

mapping trained on $\mathcal{A}$ is generalized to the evaluation set $\mathcal{E}$, whose category set $\mathcal{H}^e$ does not overlap with $\mathcal{H}^a$: $\mathcal{H}^e \cap \mathcal{H}^a = \emptyset$.

6.2.2 Web Text-Aided Attention Model

Our proposed attention model aims to learn and transfer the visual clues learned from the web text information to image contexts by the guidance of the attention-based mapping. In this work, we rely on the online web text retrieval method to acquire the web text descriptions/captions of the training images in $\mathcal{A}$ for attention network training and the test images in $\mathcal{E}$ for one-shot learning, respectively. In particular, we use the powerful reverse search engine of Google Images to retrieve the web resources of the query images. Then, we remove the non-sense components (e.g., digits, punctuation, etc) from the raw text data and feed the remaining text information to word2vec [66] model to extract the 300-dimensional word embeddings.

Semantic-Level Encoder: We present a semantic-level encoder that uses the semantic convolutional network (s-CNN) to generate the new representation of the aforementioned web text data. The s-CNN has the capability of encoding semantic correlation between the task-generic filter bank¹ and word-wise text data. However, since the s-CNN text feature vectors have different dimensionality and distribution

¹s-CNN developed two different banks of semantic filters for object-centric problems and scene-centric problems, respectively.

with the CNN visual feature vectors, the s-CNN text features are projected to a common coding space with the CNN visual features by a single layer perceptron. Suppose that we are given the web text data of a training image in $\mathcal{A}$, we apply the task-generic semantic filters on top of the word embeddings of the web text data to compute high-level text feature vector $\mathbf{z}$. By projecting $\mathbf{z}$ to the new encoding space, we obtain the semantic coding vector $\mathbf{t}$, which is written by:

$$\mathbf{t} = \mathbf{W}_t \mathbf{z} + \mathbf{b}_t, \quad (6.1)$$

where $\mathbf{W}_t \in \mathbb{R}^{d \times l}$ and $\mathbf{b}_t \in \mathbb{R}^d$ are the weight parameters to be trained. The overview of semantic feature extraction and encoding for an image exemplar is illustrated in Figure 6.1.

Visual-Level Encoder: As convolutional networks become one of the most popular modeling architectures in computer vision community, many visual recognition tasks benefit from the sweeping generalization of the mid-level CNN-based representation. This visual-level encoder uses a VGG network [48] to extract the spatial visual features off-the-shelf. Specifically, we employ the last convolutional layer of the VGG16 network to obtain 512 feature maps, each of which have a dimension of 7×7 . Considering the grid locations with features in depth, each of the k (here $k = 49$) grid locations corresponds to a 512-dimensional spatial feature vector.

In order to apply local feature encoders to $\mathcal{X}$, we design a transformation function involving the rectifier activator in the single layer perceptron. As a result, we form a set of spatial coding vectors $\mathcal{V} = [\mathbf{v}_1, \dots, \mathbf{v}_k]$, and $\mathbf{v}_i$ is computed by:

$$\mathbf{v}_i = \text{ReLU}(\mathbf{W}_s \mathbf{x}_i + \mathbf{b}_s), \quad (6.2)$$

where $\mathbf{x}_i \in \mathbb{R}^{512}$ represents a spatial feature vector at the i_{th} grid location, $\mathbf{W}_s \in \mathbb{R}^{d \times 512}$ and $\mathbf{b}_s \in \mathbb{R}^d$ are the weight parameters to be trained in the attention network. The spatial coding vectors $\mathcal{V}$ is formed as a feature matrix with a dimensionality of $d \times k$ for computation.

Besides spatial visual feature encoder, an additional visual-level encoder on the global visual feature vector $\mathcal{X}^*$ is developed in this work. By taking global average pooling on the spatial visual features, the global visual feature vector is derived

by:

$$\mathcal{X}^* = \frac{1}{k} \sum_{i=1}^k \mathbf{x}_i. \quad (6.3)$$

Then, we design a coding algorithm that has a similar transformation function as shown in Eq. (6.2) for $\mathcal{X}^*$:

$$\mathbf{v}^g = \text{ReLU}(\mathbf{W}_g \mathcal{X}^* + \mathbf{b}_g), \quad (6.4)$$

where $\mathbf{W}_g$ and $\mathbf{b}_g$ are the model parameters that have the identical dimensions to $\mathbf{W}_s$ and $\mathbf{b}_s$ in Eq. (6.2), respectively. The encoder of global visual features works in a similar manner to the supervised coding method presented in [167].

Spatial Attention-based Decoder: The idea of spatial attention model is to generate attention maps based on the semantic coding vectors for the spatial visual features. The goal is to derive the attended new image representation, which is known as the context vector $\mathbf{c}$ in attention mechanism:

$$\mathbf{c} = f(\mathcal{V}, \mathbf{t}), \quad (6.5)$$

where f is the spatial attention function that captures the relevancy of the web text information and the local region information within the images. Then, we apply a vector of attention weights $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_k]$ over the spatial coding vectors $\mathcal{V}$, which can be interpreted as a weighted pooling strategy. Based on the distribution of attention weights, the context vector is derived by:

$$\mathbf{c} = \boldsymbol{\alpha} \cdot \mathcal{V}^\top \in \mathbb{R}^d. \quad (6.6)$$

To calculate the attention weight vector $\boldsymbol{\alpha}$, we firstly take the dot product of the spatial visual coding vectors and the semantic coding vector:

$$\boldsymbol{\alpha}' = \text{score}(\mathbf{t}, \mathcal{V}) = \mathbf{t}^\top \cdot \mathcal{V}. \quad (6.7)$$

As a confidence score vector, $\boldsymbol{\alpha}' \in \mathbb{R}^k$ measures the relevancy of the local visual coding vectors and the semantic coding vectors. Here, we use dot product to compute the confidence score because [168] has experimentally shown that dot

FIGURE 6.2: An illustration of our proposed spatial attention network.

product outperforms other alternatives, such as latent space learning or multi-layer perceptron (MLP) training in the case of spatial attention modeling. Finally, the confidence score is normalized to the range of 0 to 1 by using a logistic sigmoid function:

$$\alpha = \frac{1}{1 + \exp(\alpha')} = \text{sigmoid}(\alpha'). \quad (6.8)$$

The above attention weight vector α learned from the text encoders is applied to the visual encoders to create a new representation for one-shot classification. This attention context vector is considered as the attended image representation. During network training, we apply a softmax classifier on top of the proposed attention encoder-decoder in a manner of end-to-end gradient-based learning. The overview of the spatial attention model is depicted in Figure 6.2.

6.2.3 Adaptive Attention Model

The attention-based mapping between local visual features and web text features enables us to attend to some particular local regions within the images by the aid of external knowledge. However, the risk of involving web text data should not be overlooked due to its noisy nature. To address this issue, we propose an adaptive weighting algorithm that helps the classification system determine when to rely on the attention model and when to rely on the overall visual signals.

Intuitively, a latent weight is introduced in our adaptive model to decide how much external knowledge information and global visual information is used in the system.

The advantage of involving global visual information is that it can aggregate all the local features in different regions. However, the salient image content may be distracted and overwhelmed by some interfering content and lead to the failure of image generalization. Therefore, this adaptive attention model provides attention maps guidance to the achieve more salient image representation.

As shown in Figure 6.3, a new context vector $\mathbf{c}'$ is defined as a mixture of the context vector obtained from the spatial attention model and the global visual coding vector. Hence, the new and adaptive context vector $\mathbf{c}'$ is formed as:

$$\mathbf{c}' = \beta \mathbf{c} + (1 - \beta) \mathbf{v}^g, \quad (6.9)$$

where β is the extra latent weight to be trained in attention decoder with the attention weight vector α . Specially, we train an additional element in the confidence score vector α' shown in Eq. (6.7):

$$\alpha' = \{\mathbf{t}^\top \mathcal{V}; \mathbf{t}^\top \mathbf{v}^g\}, \quad (6.10)$$

where $\{;\}$ denotes the concatenation operation, and α' is the new attention score vector. Here, $\mathbf{t}^\top \mathbf{v}^g$ is actually another confidence score that indicates the relatedness of the global visual features and web semantic features in overall. The higher this confidence score is, the more useful external knowledge involved is to the target task. With this additional confidence score, we redefine the attention weight vector (i.e., the normalization of attention score vector) in Eq. (6.8) for the adaptive mechanism as follows:

$$\alpha = \text{sigmoid}(\alpha') = \text{sigmoid}(\{\mathbf{t}^\top \mathcal{V}; \mathbf{t}^\top \mathbf{v}^g\}), \quad (6.11)$$

where $\alpha \in \mathbb{R}^{k+1}$ is the attention distribution over spatial visual coding vectors and the global visual coding vector. The latent weight scalar β is actually the last element in α of Eq. (6.11), which is normalized in the range of $[0,1]$ by the sigmoid function. Then, we feed β into Eq. (6.9) to derive the concatenated context vector $\mathbf{c}'$, which is also followed by a MLP (softmax classifier) as depicted in Figure 6.3. In general, the proposed latent weight scalar β is updated based on pairwise relatedness between different images and their corresponding semantic features. Therefore, this framework is indeed an adaptive attention network.

FIGURE 6.3: An illustration of our proposed adaptive attention network.

6.3 Experiments

In this section, we firstly introduce the datasets used in this work to evaluate one-shot classification. Then, the setup and metrics of the experiments are detailed. Lastly, the effectiveness of the proposed adaptive attention network aided by web text for image classification is validated by the performance comparison with some baseline methods and qualitative visualization of the attention maps.

6.3.1 Datasets

In this study, we work on the subsets of VOC-2007 and MIT Indoor-67 datasets we implemented in the previous chapters. The web text datasets of these two datasets used here is the same as the ones used in Chapter 5. Next, the details of these data subsets is demonstrated as follows:

VOC-2007 Subsets: The latest Pascal VOC Challenge recognizes objects from 20 visual object classes (VOC), which are divided into 4 super-classes: “*person*”, “*animal*”, “*vehicle*”, and “*indoor*”. The VOC-2007 dataset totally has 9,963 images, many of which have up to 6 objects of different classes. Thus, it is a challenging multi-label classification problem. To fairly compare the performance with other baselines, we construct two sub-datasets based on 6,295 single-label images

of VOC-2007. Both sub-datasets consist of 16 classes for network training and the rest 4 classes for one-shot learning evaluation. The evaluation set of the first dataset consists 4 classes with respect to the 4 super-classes respectively: “person”, “sheep”, “bicycle”, “chair”. The second dataset has the evaluation set of two classes from the animal category and another two classes from the vehicle category, respectively: “cat”, “dog”, “bicycle”, “motorcycle”.

MIT-Indoor67 Subset: Different from the VOC challenges that focus on object-centric images, MIT-Indoor67 involves 67 classes of indoor scene images with 5 super-classes: store, home, public spaces, leisure and working place. In general, the challenge of this scene recognition problem is the large intra-class variation across the 67 classes. To apply the one-shot learning idea to this dataset, we use 57 image classes as training data and the remaining 10 classes that do not appear in the 57 training classes for one-shot evaluation. We employ two classes from each of the super-classes to form the 10-class evaluation set, including “bakery”, “florist”, “kitchen”, “bathroom”, “airport”, “museum”, “casino”, “hair salon”, “hospital”, “kindergarten”.

To make fair comparison with the methods in the literature, we implement two different settings following the experimental settings presented in [4]. The first and simpler setting is to use the evaluation set only as the test data, from which to identify the images of the novel classes. This is the so-called “close-world” setting. The second setting is more intractable that includes both evaluation set and auxiliary set in the testing stage. In contrast to the “close-world” setting, we call it “open-world” setting in this case.

6.3.2 Experimental Setup

Multi-Attention Algorithm: In Section 6.2, we illustrate the proposed adaptive attention model with single attention scheme, where the single attention map² is generated to calculate single attended image feature vector (context vector). Furthermore, Wang et al. [4] reveal that multi-attention scheme substantially outperforms single-attention scheme for one-shot learning because multi-attention scheme can effectively avoid creating inaccurate attention maps by adding complementary

²i.e., 2D conversion of the attention weight vector α

attention map candidates. Thus, we also introduce multi-attention scheme for our adaptive attention network in our experiments. By using h sets of weight parameters $\{\mathbf{W}_t^i, \mathbf{b}_t^i\}$ in Eq. (6.1), h attention detectors $\{\mathbf{t}_i\}$ are achieved accordingly. Then, we feed the attention detectors into the adaptive attention network to acquire multiple context vectors, which are finally concatenated to form a new representation:

$$\mathbf{C} = [\mathbf{c}'_1, \dots, \mathbf{c}'_h]. \quad (6.12)$$

Implementation Details: Hyperparameters of the proposed adaptive attention model are set as follows. Adam is adopted as the optimizer for mini-batch gradient decent with learning rate of 1e-5. The momentum and weight decay are set to 0.9 and 0.999, respectively. The entire adaptive network is trained for 30 epochs with the mini-batch size of 32. We use the output from the last convolutional layer of VGG network and 300-dimensional word vectors [66] of web text as the inputs to the visual-level encoder and the semantic-level encoder, respectively. The s-CNN network for semantic-level encoder produces 200-dimensional and 365 dimensional text feature vectors for object classification and scene classification, respectively.

Metrics and Compared Methods: Like most existing metric learning methods [148, 152, 169] for one-shot learning, we take distance/similarity measure as the criterion for image retrieval or classification in this work. In particular, cosine similarity is employed to compare the pairwise similarity between the context vectors of exemplar images and other testing images in the evaluation set. Similar to the metric learning approach [4], we use images in evaluation set as exemplar images in turn and compute mean average precision (mAP) as the evaluation metric based on aforementioned cosine similarity scores. Our proposed adaptive attention network (AAN) with the aid of web text data is compared with the following baseline methods:

- *VGG-FC (VF)*: We use the pre-trained VGG network [48] as off-the-shelf deep visual feature extractor, which outputs the activations of the penultimate layer as mid-level image representation. We employ cosine distance between images as measure of similarity.
- *Zero-shot Learning (ZSL)*: Zero-shot learning is a mapping approach that tries to project the knowledge learned from known classes to the vector

space of novel class(es) via attributes [170] or text embedding (e.g., class label [171] embedding). This highly relies on the semantic correlation between the known labels and novel label(s). We implement a zero-shot learning regime [171] to predict the embedding vectors of the novel classes in network training. Unlike pairwise similarity comparison presented in [171], we measure the cosine distance between the embedding vectors of exemplar images and evaluation set in the close-world setting.

- *Supervised Coding (SC)*: We implement the supervised coding approach following [167], which firstly encodes local VGG features into vector space and then aggregates the coding vectors via sum pooling. Similar to the VF method, cosine similarity is adopted to measure distance between exemplar and evaluation images.
- *Multi-Attention (5) (MA (5))*: We implement class tag-based attention network in [4] with 5 attention maps in the specific applications, which shows the best results in the original paper. Note that, we also average the embedding of each word in the class tags as the semantic representation of the class tags if the class tags contains multiple words.

6.3.3 Results on VOC-2007 Subsets

In this section, We show the experimental results on the two subsets of VOC-2007 object classification problem in Table 6.1 (A) and Table 6.1 (B). The **VOC-2007 subset 1** consist of 4 classes from 4 different super-classes, while The **VOC-2007 subset 2** focuses on two similar classes from “*animals*” and “*vehicles*”, respectively. In overall, the mAP performance on **VOC-2007 subset 1** is not as good as that of **VOC-2007 subset 2** due to the data imbalance problem caused by including class “person”.

In the close-world setting, we observe that the zero-shot learning (ZSL) method performs the worst in both subsets. In fact, it is really difficult to imply the novel classes without any visual clues informed by the semantic hints. However, [4] has proven that the performance can be significantly improved even if only one exemplar is adopted as reference. In contrast, the VGG-FC (VF) method can represent the characteristics of the novel classes very well. It is presumably because

Method	Close	Open	Method	Close	Open
VF	76.1	58.2	VF	79.1	60.1
ZSL	33.0	-	ZSL	37.1	-
SC	72.9	47.7	SC	73.2	48.8
MA (5)	76.8	59.0	MA (5)	80.4	59.4
AAN	79.1	65.1	ANN	83.1	61.5
AAN (2)	79.3	65.4	AAN (2)	83.2	62.4
AAN (5)	79.0	65.3	AAN (5)	82.6	62.1

(A)
(B)

TABLE 6.1: Overall mAP (%) comparison of the proposed adaptive attention network to the other baseline methods on VOC-2007: (A) **subset 1**, (B) **subset 2** The number in the parenthesis indicates the number of attention maps involved. The last two columns in the tables represent the close-world and open-world experimental settings, respectively.

the 1000-class ImageNet is used for the VGG network training, which somehow overlaps with the image content of VOC challenges. Similar to the proposed adaptive attention network (AAN), both the supervised coding method (SC) and class tag-based multi-attention model (MA (5)) employ local VGG feature aggregation. Considering the external knowledge that web text brings about, our proposed AAN networks can significantly boost the one-shot classification in both close-world and more challenging open-world settings. From Table 6.1 (A) and Table 6.1 (B), it is interestingly observed that increasing number of attention maps from 2 to 5 for the proposed AAN methods cannot capture more useful information in VOC-2007 object classification.

6.3.4 Results on MIT-Indoor67 Subset

Besides object classification problem, we also validate the effectiveness of our proposed method on MIT-Indoor67 scene classification, and the results are shown in Table 6.2. Compared to the object classification results, we observe that the overall performance of all the listed methods on the indoor scene dataset degrades, especially the class tag-based attention (MA) model [4]. Furthermore, it discloses the applicability limitation of the MA model for different one-shot recognition problems. In MIT-Indoor67 dataset, for example, there are three different class tags “buffet”, “fastfood”, and “restaurant” under the same super-class “leisure”. Since

TABLE 6.2: Overall mAP (%) comparison of the proposed adaptive attention network to the other baseline methods on MIT-Indoor67 subset.

Method	Close	Open
VF	64.6	32.1
ZSL	25.1	-
SC	55.2	20.7
MA (5)	58.4	25.9
AAN	64.3	31.6
AAN (2)	65.9	31.9
AAN (5)	66.4	32.8

the cosine similarity scores between any two out of three are very high, it is intractable to attend to the discriminative local regions using class tag information. Different from the MA method, our proposed adaptive attention model is aided by more informative web text descriptions than using class tag alone. This means the AAN has better generalization for different recognition problems, including object and scene classification. With the increase of attention maps, we observe significant advantage of the AAN method presented in this work over the other methods in Table 6.2.

6.3.5 Attention Maps Visualization

To qualitatively validate the effectiveness of the proposed adaptive attention network, we visualize the attention maps generated on the novel classes $\mathcal{H}^e$ of the evaluation set $\mathcal{E}$. Some example results are shown in Figure 6.4 and Figure 6.5 for the **VOC-2007 Subset 2**, where two attention maps are generated by multi-attention algorithm for each exemplar image and $\mathcal{H}^e = \{\text{“cat”}, \text{“dog”}, \text{“bicycle”}, \text{“motorcycle”}\}$ does not appear in the network training for attention map generation. Both of the two methods can highlight the class-related regions within the image even these classes are not seen during network training stage. However, we observe that the class tag-based MA method [4] cannot help to capture the indicative content in some classes, such as “bicycle” and “motorbike” examples in Figure 6.4. This means the MA methods fails to map the knowledge learned from training images of the seen classes in “*vehicle*” to the novel classes. By contrast to the MA method, our proposed adaptive attention model is able to identify the regions of interest for “bicycle” and “motorbike” examples using more informative web text. Furthermore, we also highlight the values of latent weight scalar β

FIGURE 6.4: Visualization of attention maps of MA method [4] on the VOC subset 2 novel classes, where the intensity indicates the activeness of different regions.

learned by the adaptive attention mechanism in the Figure 6.5. The higher the values of β are, the more attention-based guidance will be used in image recognition, and vice versa. Notably, we can see that the weight scalars values are truly adaptive to different observations, which determines how much attention should be paid to visual clues learned from the web text.

6.4 Conclusion

In this chapter, we propose an adaptive attention encoder-decoder network with the aid of web text descriptions for one-shot learning. In overall, this implementation is actually the extension work of the web text-aided image classification in a specific application. Unlike the previous multimodal attention network for one-shot learning, we introduce more detailed web text information to describe the

FIGURE 6.5: Visualization of attention maps of our proposed adaptive attention model on the novel classes, where the β values (shown as “b”) are displayed at the upper left corners of the attention maps.

image content and this can lead to superb attention maps in performance. Furthermore, we develop an adaptive mechanism to deal with the noisy nature of web unstructured data. Experimental results show that the web text-aided adaptive attention models outperform the comparing methods in both object and indoor scene classification problems.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we have improved image classification by aid of heterogeneous data from heterogeneous sources through the online web resources search. From the aspect of the decision level and feature level, we incorporate web text modality into image modality for classification. The challenge in this study is how to handle the diversity of representation, distribution and density of multiple modalities from cross-domain data. Throughout Chapter 2, the related work for text representation learning, image representation learning are reviewed, which lead to the fusion of multiple modalities to aid image classification. Our work presented in this thesis is inspired by these models.

In Chapter 3, we proposed an adaptive decision-level fusion algorithm for the integration of the visual feature-based and web textual feature-based classification results. Two separate classifiers based on visual and textual features are first built, and the decision-level fusion is then performed by applying pairwise adaptive weight vectors to image-based and text-based classification scores. The goal of handling different modality of data separately is to reduce the vulnerable interaction of heterogeneous data. This adaptive decision-level fusion system preserves the information of multimodal structure, whereas it dismisses the underlying correlations across multiple modalities.

In Chapter 4, we introduce a feature-level multimodal fusion framework that uses an unified neural network to learn a shared multimodal representation, which preserves both the information of multimodal structure and the correlation between multimodal feature representations. On the image side, training images are represented by visual features, which are obtained from the pre-trained CNNs such as VGG [47]. On the text hand, web text data is described by a newly proposed semantic matching neural network (SMNN). By learning the correlations across bimodality through mid-level feature space, we fuse the visual features extracted from training data and text features extracted from web text data on the feature level using neural network training.

However, a potential limitation of the SMNN model is the use of semantic filters that derived from the reference documents with respect to the task-specific class labels. This limitation may affect the applicability of our proposed feature-level fusion method. To this end, we also investigate another semantic neural network so-called semantic convolutional neural network (s-CNN) in Chapter 5. Our s-CNN model employs task-generic semantic filters in a convolutional network architecture, which is incorporated into the image-based CNN models in an end-to-end framework.

Different from previous chapters that focus on general and conventional web text-aided image classification, we extend the application of s-CNN text representation in more challenging one-shot image classification problems in Chapter 6. In this web-aided one-shot learning model, we adopt an adaptive attention network to effectively extract the transferable visual clues from the web text-aided image representation, which can be used to recognize unfamiliar images from a single glimpse with very little human-annotated information.

In summary, the aid of web text data for image classification involves two modalities of data from heterogeneous sources that can be regarded as different views on the image modality data. According to different types of feature representation, web text can always contribute to the image classification from different perspectives. For example, web text data can also help to learn multimodal attention models or produce more salient and semantic representations for enhanced video recognition.

7.2 Future Directions

Recently, multimedia information fusion is playing an increasing important role in computer vision and artificial intelligence. We plan to focus on the applications of multimodal data fusion methods. The future research related to the work presented in this thesis and multimodal fusion are summarized as:

- 1 **Enrich word vector representation:** As a core representation technique of NLP, word embedding has the same wide-ranging impact on NLP as pre-trained ImageNet models had on computer vision. In spite of pre-trained word vectors that learn general-purpose text features, good language understanding should also need fine-tuning on the existed embeddings. Some exciting methods including Embeddings from Language Models (ELMo) [172], Universal Language Model Fine-tuning (ULMFiT) [173], and most recently BERT [174] transformer models have been proposed to make word vector representation learning more comprehensive. To some extent, they have brought the NLP community really close to having an “ImageNet for language”. In the future, our exploration on text representation will mainly rely on these higher-level language models with post-fining techniques. Our next step is to implement BERT model as a pre-trained language model for the object-centric and scene-centric scenarios, on which our proposed SMNN and s-CNN models can be built to learn more overwhelming text representations.
- 2 **Exploit comprehensive adaptive mechanism for noise cancelling:** Although the web text data associated with images is rich of visual clues, it still cannot get rid of the negative influence of noise. Therefore, we plan to build adaptive mechanism to selectively use web text data, making the framework more robust and immune to the interference of noisy components in the web resources.
- 3 **More challenging image classification tasks:** Considering the advantage of the online web resources search and the extension in one-shot learning discussed in this thesis. We have reasons to believe that the web text-aided framework can be used as a potential protocol to classify instances of completely unseen visual class with no training data available, which is referred to as zero-shot learning [171]. In addition, web-supervised video classification

is also an interesting research topic of visual recognition. For example, Gan et al. [144, 145] use web images and videos to train deep neural networks for video recognition. In the future, we may explore multimodal attention models to learn more salient and semantic representations for image classification (with CNN architectures) or video classification (with RNN architectures), which explore the visually attentive image regions with the aid of web text.

Author's Publications

Journal Articles

- **Dongzhe Wang**, Kezhi Mao, “Task-Generic Semantic Convolutional Neural Network for Web Text-Aided Image Classification,” *Neurocomputing* 329 (2019): 103-115.
- **Dongzhe Wang**, Kezhi Mao, “Learning Semantic Text Features for Web Text-Aided Image Classification,” *Revised for IEEE Transactions on Multimedia*.
- **Dongzhe Wang**, Kezhi Mao, “Adaptive Web Text-Aided Attention Network for One-Shot Image Classification,” *Submitted to Pattern Recognition*.

Conference Proceedings

- **Dongzhe Wang**, Kezhi Mao, “Multimodal Object Classification using Bidirectional Gated Recurrent Unit Networks,” *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, IEEE, 2018.
- **Dongzhe Wang**, Kezhi Mao, and Gee-Wah Ng, “Convolutional Neural Networks and Multimodal Fusion for Text Aided Image Classification,” *Information Fusion (Fusion)*, 2017 20th International Conference on, IEEE, 2017.
- **Dongzhe Wang**, Kezhi Mao, Gee-Wah Ng, and Tien Pham, “Adaptive Multimodal Fusion with Web Resources for Scene Classification,” *Information Fusion (FUSION)*, 2016 19th International Conference on, IEEE, 2016.

- **Dongzhe Wang**, Rui Zhao, and Kezhi Mao, “Partially Connected ELM for Fast and Effective Scene Classification,” *Proceedings of ELM-2015 Volume 2*, Springer, 2016.
- **Dongzhe Wang**, Kezhi Mao, and Gee-Wah Ng, “Improving Scene Classification by Fusion of Training Data and Web Resources,” *Information Fusion (FUSION), 2015 18th International Conference on*, IEEE, 2015.

Bibliography

- [1] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.
- [2] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, 2014.
- [3] Dongzhe Wang, Kezhi Mao, and Gee-Wah Ng. Convolutional neural networks and multimodal fusion for text aided image classification. In *Information Fusion (FUSION), International Conference on*, 2017.
- [4] Peng Wang, Lingqiao Liu, Chunhua Shen, Zi Huang, Anton van den Hengel, and Heng Tao Shen. Multi-attention network for one shot learning. In *CVPR*, pages 22–25, 2017.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE, 2009.
- [6] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, pages 759–766. ACM, 2007.
- [7] Roi Reichart and Ari Rappoport. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *ACL*, volume 7, pages 616–623, 2007.
- [8] Xiaojin Zhu. Semi-supervised learning literature survey. 2005.
- [9] Gang Wang, Derek Hoiem, and David Forsyth. Building text features for object image classification. In *CVPR*, pages 1367–1374. IEEE, 2009.

- [10] Lexiao Tian, Dequan Zheng, and Conghui Zhu. Image classification based on the combination of text features and visual features. *International Journal of Intelligent Systems*, 28(3):242–256, 2013.
- [11] Amel Znaidia, Aymen Shabou, Adrian Popescu, Hervé Le Borgne, and Céline Hudelot. Multimodal feature generation framework for semantic image classification. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, page 38. ACM, 2012.
- [12] Shenghua Gao, Zhengxiang Wang, Liang-Tien Chia, and Ivor Wai-Hung Tsang. Automatic image tagging via category label and web data. In *Proceedings of the international conference on Multimedia*, pages 1115–1118. ACM, 2010.
- [13] Dongzhe Wang, Kezhi Mao, Gee-Wah Ng, and Tien Pham. Adaptive multimodal fusion with web resources for scene classification. In *Information Fusion (FUSION), 2016 19th International Conference on*. IEEE, 2016.
- [14] Dongzhe Wang and Kezhi Mao. Learning semantic text features for web text aided image classification. *Revised for IEEE Transactions on Multimedia*, 2018.
- [15] Dongzhe Wang and Kezhi Mao. Task-generic semantic convolutional neural-network for web text-aided image classification. *Revised for Neurocomputing*, 2018.
- [16] Dongzhe Wang and Kezhi Mao. Adaptive web text-aided attention network for one-shot image classification. *Submitted to International Conference on Multimedia Modeling (Under Review)*, 2019.
- [17] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2, 2004.
- [18] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–492. University of California Press, 1951.
- [19] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157. Ieee, 1999.

- [20] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, volume 2, pages 2169–2178. IEEE, 2006.
- [21] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*. IEEE, 2005.
- [22] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *Proc. ICCV*, 2007.
- [23] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367. IEEE, 2010.
- [24] Shenghua Gao, Ivor Waihung Tsang, Liang-Tien Chia, and Peilin Zhao. Local features are not lonely—laplacian sparse coding for image classification. In *CVPR*, pages 3555–3561. IEEE, 2010.
- [25] Chunjie Zhang, Jing Liu, Qi Tian, Changsheng Xu, Hanqing Lu, and Songde Ma. Image classification by non-negative sparse coding, low-rank and sparse decomposition. In *CVPR*, pages 1673–1680. IEEE, 2011.
- [26] Chunjie Zhang, Jing Liu, Chao Liang, Zhe Xue, Junbiao Pang, and Qingming Huang. Image classification by non-negative sparse coding, correlation constrained low-rank and sparse decomposition. *Computer Vision and Image Understanding*, 123:14–22, 2014.
- [27] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. *Computer Vision–ECCV 2010*, pages 143–156, 2010.
- [28] Josip Krapac, Jakob Verbeek, and Frédéric Jurie. Modeling spatial layout with fisher vectors for image categorization. In *ICCV*, pages 1487–1494. IEEE, 2011.
- [29] Li-Jia Li, Hao Su, Yongwhan Lim, and Li Fei-Fei. Objects as attributes for scene classification. In *Trends and Topics in Computer Vision*, pages 57–69. Springer, 2012.

- [30] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.
- [31] 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.
- [32] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [33] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.
- [34] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.
- [35] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.
- [36] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [37] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.
- [38] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.
- [39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.
- [40] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017.

- [41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep neural networks. In *NIPS*, pages 1097–1105, 2012.
- [42] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
- [43] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Antonio Torralba, and Aude Oliva. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*, 2016.
- [44] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *TPAMI*, 35(12):2891–2903, 2013.
- [45] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137, 2015.
- [46] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.
- [47] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.
- [48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [49] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.
- [50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [51] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.
- [52] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *NIPS*, pages 3856–3866, 2017.

- [53] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. 2014.
- [54] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724. IEEE, 2014.
- [55] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013.
- [56] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.
- [57] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.
- [58] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.
- [59] Ian Jolliffe. Principal component analysis. In *International encyclopedia of statistical science*, pages 1094–1096. Springer, 2011.
- [60] Ye Zhang and Byron Wallace. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. *Computer Science*, 2015.
- [61] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In *NIPS*, pages 2042–2050, 2014.
- [62] Cícero Nogueira dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.
- [63] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. *Eprint Arxiv*, 1, 2014.

- [64] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. *CoRR*, abs/1412.1058, 2014.
- [65] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Computer Science*, 2013.
- [66] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
- [67] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.
- [68] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, pages 1137–1155, 2003.
- [69] Rie Johnson and Tong Zhang. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2014.
- [70] Rie Johnson and Tong Zhang. Semi-supervised convolutional neural networks for text categorization via region embedding. In *NIPS*, pages 919–927, 2015.
- [71] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.
- [72] Saratha Sathasivam and Wan Ahmad Tajuddin Wan Abdullah. Logic learning in hopfield networks. *arXiv preprint arXiv:0804.4075*, 2008.
- [73] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [74] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014.

- [75] Amr El-Desoky Mousa and Björn Schuller. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *Proceedings EACL*, 2017.
- [76] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.
- [77] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 207–212, 2016.
- [78] Dongzhe Wang and Kezhi Mao. Multimodal object classification using bidirectional gated recurrent unit networks. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018.
- [79] Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *ACM SIGKDD*, pages 797–806. ACM, 2009.
- [80] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.
- [81] Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000.
- [82] Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai, and Yong Yu. Heterogeneous transfer learning for image clustering via the social web. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 1–9. Association for Computational Linguistics, 2009.

- [83] Thi Quynh Nhi Tran, Hervé Le Borgne, and Michel Crucianu. Cross-modal classification by completing unimodal representations. In *Proceedings of the 2016 ACM Workshop on Vision and Language Integration Meets Multimedia Fusion*, pages 17–25. ACM, 2016.
- [84] Yunchao Wei, Yao Zhao, Canyi Lu, Shikui Wei, Luoqi Liu, Zhenfeng Zhu, and Shuicheng Yan. Cross-modal retrieval with cnn visual features: A new baseline. *IEEE transactions on cybernetics*, 47(2):449–460, 2017.
- [85] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.
- [86] Shotaro Akaho. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.
- [87] Malte Kuss and Thore Graepel. The geometry of kernel canonical correlation analysis. 2003.
- [88] Thi Quynh Nhi Tran, Hervé Le Borgne, and Michel Crucianu. Cross-modal classification by completing unimodal representations. In *Proceedings of the 2016 ACM workshop on Vision and Language Integration Meets Multimedia Fusion*, pages 17–25. ACM, 2016.
- [89] Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *AAAI*, 2011.
- [90] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.
- [91] Qiang Zhu, Mei-Chen Yeh, and Kwang-Ting Cheng. Multimodal fusion using learned text concepts for image categorization. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 211–220. ACM, 2006.
- [92] Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Multimodal semi-supervised learning for image classification. In *CVPR 2010-23rd IEEE Conference on Computer Vision & Pattern Recognition*, pages 902–909. IEEE Computer Society, 2010.

-
- [93] Soujanya Poria, Haiyun Peng, Amir Hussain, Newton Howard, and Erik Cambria. Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing*, 2017.
 - [94] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, pages 2222–2230, 2012.
 - [95] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L Yuille. Explain images with multimodal recurrent neural networks. *arXiv preprint arXiv:1410.1090*, 2014.
 - [96] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.
 - [97] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.
 - [98] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015.
 - [99] Xinlei Chen and C Lawrence Zitnick. Mind’s eye: A recurrent visual representation for image caption generation. In *CVPR*, pages 2422–2431, 2015.
 - [100] Lin Ma, Zhengdong Lu, Lifeng Shang, and Hang Li. Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2623–2631, 2015.
 - [101] Lin Ma, Zhengdong Lu, and Hang Li. Learning to answer questions from image using convolutional neural network. *arXiv preprint arXiv:1506.00333*, 2015.
 - [102] Vedran Vukotić, Christian Raymond, and Guillaume Gravier. Multimodal and crossmodal representation learning from textual and visual features with bidirectional deep neural networks for video hyperlinking. In *Proceedings of*

- the 2016 ACM Workshop on Vision and Language Integration Meets Multimedia Fusion*, pages 37–44, 2016.
- [103] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.
- [104] Betsy Sparrow, Jenny Liu, and Daniel M Wegner. Google effects on memory: Cognitive consequences of having information at our fingertips. *science*, 333(6043):776–778, 2011.
- [105] Dongzhe Wang, Kezhi Mao, and Gee-Wah Ng. Improving scene classification by fusion of training data and web resources. In *Information Fusion (FUSION), 2015 18th International Conference on*. IEEE, 2015.
- [106] Tomokazu Ohshiro, Dora E Angelaki, and Gregory C DeAngelis. A normalization model of multisensory integration. *Nature neuroscience*, 14(6):775–782, 2011.
- [107] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [108] Li-Jia Li, Hao Su, Li Fei-Fei, and Eric P Xing. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, pages 1378–1386, 2010.
- [109] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801. IEEE, 2009.
- [110] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.
- [111] Li-Jia Li and Li Fei-Fei. What, where and who? classifying events by scene and object recognition. In *ICCV*, pages 1–8. IEEE, 2007.
- [112] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.
- [113] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes

- Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [114] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, 106(1):59–70, 2007.
- [115] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [116] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [117] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.
- [118] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [119] Dong Seon Cheng, Francesco Setti, Nicola Zeni, Roberta Ferrario, and Marco Cristani. Semantically-driven automatic creation of training sets for object recognition. *Computer Vision and Image Understanding*, 131:56–71, 2015.
- [120] Jiawei Chen, Yin Cui, Guangnan Ye, Dong Liu, and Shih-Fu Chang. Event-driven semantic concept discovery by exploiting weakly tagged internet images. In *Proceedings of International Conference on Multimedia Retrieval*, page 1. ACM, 2014.
- [121] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.
- [122] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

- [123] KZ Mao. Rbf neural network center selection based on fisher ratio class separability measure. *IEEE Transactions on Neural Networks*, 13(5):1211–1217, 2002.
- [124] Ke Zhi Mao and Guang-Bin Huang. Neuron selection for rbf neural network classifier based on data structure preserving criterion. *IEEE Transactions on Neural Networks*, 16(6):1531–1540, 2005.
- [125] François Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, 2016.
- [126] Thibaut Durand, Nicolas Thome, and Matthieu Cord. Weldon: Weakly supervised learning of deep convolutional neural networks. In *CVPR*, pages 4743–4752, 2016.
- [127] Zhen Zuo, Gang Wang, Bing Shuai, Lifan Zhao, and Qingxiong Yang. Exemplar based deep discriminative and shareable feature learning for scene image classification. *Pattern Recognition*, 48(10):3004–3015, 2015.
- [128] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.
- [129] Yunchen Pu, Win Yuan, Andrew Stevens, Chunyuan Li, and Lawrence Carin. A deep generative deconvolutional image model. In *Artificial Intelligence and Statistics*, pages 741–750, 2016.
- [130] Shenghua Gao, Lixin Duan, and Ivor W Tsang. Defeatneta deep conventional image representation for image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(3):494–505, 2016.
- [131] Di Lin, Cewu Lu, Renjie Liao, and Jiaya Jia. Learning important spatial pooling regions for scene classification. In *CVPR*, pages 3726–3733, 2014.
- [132] Markus Koskela and Jorma Laaksonen. Convolutional network features for scene recognition. In *Proceedings of the ACM International Conference on Multimedia*, pages 1169–1172. ACM, 2014.
- [133] Ilja Kuzborskij, Fabio Maria Carlucci, and Barbara Caputo. When naive bayes nearest neighbors meet convolutional neural networks. In *CVPR*, pages 2100–2109, 2016.

- [134] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. Hcp: A flexible cnn framework for multi-label image classification. *TPAMI*, 38(9):1901–1907, 2016.
- [135] Xiaojin Zhu. Semi-supervised learning. In *Encyclopedia of machine learning*, pages 892–897. Springer, 2011.
- [136] Rui Zhao and Kezhi Mao. Fuzzy bag-of-words model for document representation. *IEEE Transactions on Fuzzy Systems*, 2017.
- [137] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, pages 160–167. ACM, 2008.
- [138] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, pages 413–420, 2009.
- [139] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1): 98–136, 2015.
- [140] Ilja Kuzborskij, Fabio Maria Carlucci, and Barbara Caputo. When naïve bayes nearest neighbours meet convolutional neural networks. *CoRR*, abs/1511.03853, 2015.
- [141] Chaoqun Weng, Hongxing Wang, Junsong Yuan, and Xudong Jiang. Discovering class-specific spatial layouts for scene recognition. *IEEE Signal Processing Letters*, 2016.
- [142] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *CVPR Workshops*, pages 512–519. IEEE, 2014.
- [143] Yunchao Wei, Wei Xia, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. Cnn: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*, 2014.

- [144] C. Gan, T. Yao, K. Yang, Y. Yang, and T. Mei. You lead, we exceed: Labor-free video concept learning by jointly exploiting web videos and images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 923–932, 2016. doi: 10.1109/CVPR.2016.106.
- [145] Chuang Gan, Chen Sun, Lixin Duan, and Boqing Gong. Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In *European Conference on Computer Vision (ECCV)*, 2016.
- [146] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *TPAMI*, 28(4):594–611, 2006.
- [147] Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, 2011.
- [148] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.
- [149] Joseph J Lim, Ruslan R Salakhutdinov, and Antonio Torralba. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, pages 118–126, 2011.
- [150] Li Fe-Fei et al. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, pages 1134–1141, 2003.
- [151] Ruslan Salakhutdinov, Joshua Tenenbaum, and Antonio Torralba. One-shot learning with a hierarchical nonparametric bayesian model. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 195–206, 2012.
- [152] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.
- [153] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.

- [154] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, pages 1842–1850, 2016.
- [155] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, pages 4077–4087, 2017.
- [156] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *NIPS*, pages 2204–2212, 2014.
- [157] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*, pages 2956–2964, 2015.
- [158] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, 2017.
- [159] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.
- [160] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, volume 6, page 2, 2017.
- [161] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.
- [162] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015.
- [163] Yichuan Tang, Nitish Srivastava, and Ruslan R Salakhutdinov. Learning generative models with visual attention. In *NIPS*, pages 1808–1816, 2014.
- [164] Jongwon Choi, Hyung Jin Chang, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. Visual tracking using attention-modulated disintegration and integration. In *ICCV*, pages 4321–4330, 2016.

- [165] Jason Kuen, Zhenhua Wang, and Gang Wang. Recurrent attentional networks for saliency detection. In *ICCV*, pages 3668–3677, 2016.
- [166] John E Laird, Keegan R Kinkade, Shiwali Mohan, and Joseph Z Xu. Cognitive robotics using the soar cognitive architecture. In *Workshops at AAAI on Artificial Intelligence, Cognitive Robotics*, 2012.
- [167] Lingqiao Liu, Peng Wang, Chunhua Shen, Lei Wang, Anton Van Den Hengel, Chao Wang, and Heng Tao Shen. Compositional model based fisher vector coding for image classification. *TPAMI*, 39(12):2335–2348, 2017.
- [168] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2015.
- [169] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, volume 1, pages 539–546, 2005.
- [170] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009.
- [171] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013.
- [172] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, 2018.
- [173] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. In *Proceedings of ACL*, 2018.
- [174] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.