Feature learning for RGB-D scene understanding

Wang, Anran

2016

Wang, A. (2016). Feature learning for RGB-D scene understanding. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/68538>

<https://doi.org/10.32657/10356/68538>

Feature Learning for RGB-D Scene Understanding

A thesis submitted
for a degree of Doctor of Philosophy
by

Anran WANG

School of Computer Science and Engineering
Nanyang Technological University

2016

Abstract

Scene understanding is an important and fundamental problem in computer vision and is critical in applications of robotics and augmented reality. Scene understanding includes many tasks such as scene labeling, object recognition and scene classification. Most previous scene understanding methods focus on outdoor scenes. In contrast, indoor scene understanding is more challenging, due to poor illumination and cluttered objects. With the wide availability of affordable RGB-D cameras such as Kinect, huge changes have been made to indoor scene analysis due to the rich 3D geometry information provided by depth measurements.

Feature extraction is the key part for scene understanding tasks. Most of the early methods extract hand-crafted features. However, the performance of such feature extractors highly depends on variations in hand-crafting and combinations. The designing process requires empirical understanding of data, thus hard to systematically extend to different modalities. In addition, the hand-crafted features usually capture a subset of recognition cues from raw data, which might ignore some useful information. Thus, in this research, we focus on feature learning with raw data as input.

Particularly, we explore feature learning on three different tasks of indoor scene understanding using RGB-D input:

- Scene labeling: The aim is to densely assign a category label (e.g. table, TV) to each pixel in an image. Inspired by the success of unsupervised feature learning, we start by adapting the existing unsupervised feature learning technique to directly learn features from RGB-D images. Typically, better performance could be achieved by further applying feature encoding over the learned features to build

”bag of words” type of features. However, feature learning and feature encoding are performed separately, which may result in suboptimal solution. We propose to jointly optimize these two processes to derive more discriminative features.

- **Object recognition:** Most of the feature learning methods for RGB-D object recognition either learn the features for individual modalities independently, or treat RGB-D simply as undifferentiated four-channel data, which cannot adequately exploit the complementary relationship between the two modalities. To address the above issues, we propose a general *Convolutional Neural Networks* (CNN) based multi-modal learning method for RGB-D object recognition. Our multi-modal layer is designed to not only discover the most discriminative features for each modality, but also harness the complementary relationship between the two modalities.
- **Scene classification:** Methods for scene classification task to leverage local information share a similar pipeline: first densely extracting CNN features from different locations and scales of an image, and then using an encoding method. However, for state-of-the-art feature encoding techniques such as *Fisher Vector* (FV), since their components in *Gaussian Mixture Model* (GMM) are derived from densely sampled local features, many components are likely to be noisy and not informative. Such noisy property of local features has not been well considered in the existing works. Further considering the FV features from different modalities, we propose a modality and component aware feature fusion framework for RGB-D scene classification.

In this thesis, various experiments have been constructed to evaluate the performance of the proposed techniques in comparison to the state-of-the-art methods on different RGB-D databases. Encouraging results show that the proposed techniques significantly boost the performance in the studied scene understanding tasks.

Acknowledgments

I am grateful to all those who have helped and supported me throughout my PhD life.

It gives me immense pleasure to thank my supervisor, Prof. Jianfei Cai, for his persistent guidance during my graduate study. His patience and meticulous suggestions helped me overcome many difficulties and finish this thesis. Besides the valuable research skills that I learnt from him, he also affects me to be a kind and self-motivated person. Many thanks to Prof. Tat-Jen Cham, for his precious advice and guidance to my research. His co-supervision is very important for my PhD study. Also I gratefully acknowledge the help of Dr. Jiwen Lu, who gives me a lot of constructive suggestions to my research.

I also want to thank all my friends who support me a lot by providing valuable feedback to my research and sharing joyful moments with me. This report would not be possible in this form without them. In particular, thanks go to Mr. Hongyuan Zhu, Mr. Chongyu Chen, Mr. Di Xu, Mr. Fuwen Tan, Mr. Yu Guo, Ms. Mengyao Zhao, Mr. Teng Deng, Ms. Hui Li, Ms. Tianyi Zhang, Ms. Xiaoqi Yan, Ms. Min Meng, Ms. Peng Cheng, Mr. Chi-Fu Lai, Ms. Huijing Zhan, Mr. Bingbing Li, Mr. Fubing Mao, Mr. Tianyi Zhou, Mr. Yu Zhang, and Mr. Hao Yang.

Last but not least, I want to thank my family for their love during the study with the weekly video call. And I would also like to express my appreciation to my boyfriend, who is also my best friend, Mr. Xiaoning Wang, for his consistent encouragement, love, and always being by my side.

Publications

Journal papers:

Anran Wang, Jiwen Lu, Jianfei Cai, Gang Wang, Tat-Jen Cham. Unsupervised Joint Feature Learning and Encoding for RGB-D Scene Labeling. *IEEE Transactions on Image Processing*, Vol. 24, No. 11, pp. 4459-4473, 2015

Anran Wang, Jiwen Lu, Jianfei Cai, Tat-Jen Cham, Gang Wang. Large-Margin Multi-Modal Deep Learning for RGB-D Object Recognition. *IEEE Transactions on Multimedia*, Vol. 17, No. 11, pp. 1887-1898, 2015

Conference papers:

Anran Wang, Jianfei Cai, Jiwen Lu, Tat-Jen Cham. Modality and Component Aware Feature Fusion for RGB-D Scene Classification. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Anran Wang, Jianfei Cai, Jiwen Lu, Tat-Jen Cham. MMSS: Multi-modal Sharable and Specific Feature Learning for RGB-D Object Recognition. *IEEE International Conference on Computer Vision (ICCV)*, 2015.

Anran Wang, Jiwen Lu, Gang Wang, Jianfei Cai, Tat-Jen Cham. Multi-modal Unsupervised Feature Learning for RGB-D Scene Labeling. *European Conference on Computer Vision (ECCV)*, 2014.

Anran Wang, Jiwen Lu, Jianfei Cai, Gang Wang, Tat-Jen Cham. Estimating Spatial Layout of Rooms from RGB-D Videos. *International Workshop on Multimedia Signal Processing (MMSP)*, 2014.

Anran Wang, Hongyuan Zhu, Jianfei Cai, Jianxin Wu. Multi-class Cosegmentation with Pairwise Active Learning. *Pacific-Rim Conference on Multimedia (PCM)*, 2013.

Contents

Abstract	i
Acknowledgments	iii
Publications	v
List of Figures	xi
List of Tables	xv
1 Introduction	1
1.1 Background	1
1.2 Objective and Research Scope	3
1.2.1 Scene Labeling	3
1.2.2 Object Recognition	4
1.2.3 Scene Classification	5
1.3 Summary of Contributions	6
1.4 Thesis Organization	8
2 Literature Review	9
2.1 Scene Labeling	9
2.2 Object Recognition	12
2.3 Scene Classification	13
2.4 Feature Learning	15
3 Unsupervised Joint Feature Learning and Encoding for RGB-D Scene Labeling	19
3.1 Joint feature learning and encoding	20
3.1.1 Single-layer feature learning structure	22

3.1.2	Optimization Process	23
3.1.3	Hierarchical feature learning	26
3.1.4	Application on RGB-D scene labeling	26
3.2	Joint deep feature learning and encoding	28
3.2.1	Formulation	29
3.2.2	Optimization	31
3.3	Experiments	33
3.3.1	Dataset	33
3.3.2	Implementation details	33
3.3.3	Results	38
3.4	Summary	44
4	MMSS: Multi-modal Sharable and Specific Feature Learning for RGB-D	
	Object Recognition	49
4.1	Proposed Approach	51
4.1.1	Conventional CNN-based Learning Structure	51
4.1.2	Proposed Multi-modal Learning Structure	52
4.1.3	Formulation	54
4.1.4	Alternating Optimization	55
4.2	Experiments	58
4.2.1	Datasets and Experiment Setup	59
4.2.2	Results on RGB-D Object Dataset	60
4.2.3	Results on 2D3D Dataset	65
4.2.4	Parameter Analysis	67
4.3	Summary	68
5	Modality and Component Aware Feature Fusion for RGB-D Scene Classification	71
5.1	Multi-modal Proposal-based Global Feature Representation	73
5.2	Modality and Component Aware Feature Fusion	75
5.2.1	Formulation	75
5.2.2	Optimization	77

5.3	Experiments	79
5.3.1	Datasets and Experimental Setup	80
5.3.2	Results on SUNRGBD Dataset	82
5.3.3	Results on NYU Depth Dataset V2	85
5.4	Summary	88
6	Conclusions and Future Work	89
6.1	Conclusion	89
6.2	Future Work	90
	References	93

List of Figures

1.1	Understanding indoor scene is an easy task for people. Semantic information of objects and the whole scene could be inferred.	2
3.1	Our framework for RGB-D indoor scene labeling. Our method learns features from raw RGB-D input with two-layer stacking structure. Features of the two layers are concatenated to train linear SVMs over superpixels for labeling task.	20
3.2	Detailed illustration of feature extraction of the first layer: $s \times s$ color and depth patches are flattened into vectors X and Y . Z is the input vector obtained by concatenating X and Y . With learned filter matrix W and dictionary U , the sparse encoding coefficients V can be derived, which represents the feature of a $s \times s$ patch. By concatenating the features of $(S - s + 1)^2 s \times s$ small patches, we get the feature of a $S \times S$ big patch. .	21
3.3	Left: the unsupervised learning structure of the first layer. Right: the second layer structure. F_1 is the first-layer feature. F_2 is the second-layer feature.	24
3.4	Illustration of one single-layer structure in the proposed nonlinear framework. Given raw data of an RGB-D patch, we map it with several nonlinear sub-layers (here we have 2 sub-layers): each sub-layer contains linear projection matrix W , bias b and a nonlinear activation function. With the constraints of feature encoding (sparse coding), and the reconstruction constraints for each sub-layer (reconstruction and regularization of bias b), better dictionary and feature mapping could be learned. .	30
3.5	The confusion matrices of: (a) our results with one-layer JFLE structure; (b) our results with two-layer JFLE structure.	34

3.6	Visual comparisons of our JFLE and JDFLE results. From left to right: original color images, the results of combining our JFLE features and Ren’s features, the results of combining our JDFLE features and Ren’s features, ground truth.	37
3.7	Scene labeling accuracy under different λ values with single-layer JD-FLE structure on NYU depth dataset V1.	43
3.8	15 example results on NYU depth dataset V1. Rows 1st, 4th and 7th: color images. Rows 2nd, 5th and 8th: the results of combining our JDFLE features and Ren’s features. Rows 3rd, 6th and 9th: ground truth. Note that since we focus on feature learning, we did not use CRFs to smooth the labels. So the results might look a bit noisy.	46
3.9	12 example results on NYU depth dataset V2. Rows 1st and 4th: color images. Rows 2nd and 5th: the results of our JDFLE features. Rows 3rd and 6th: ground truth.	47
4.1	Illustration of our proposed multi-modal feature learning framework. The inputs X and Y are the activations of the second fully connected layers of the CNNs pre-trained on color and depth separately. The inputs are transformed by W_1 and W_2 respectively, and the transformed features T_1 and T_2 are enforced to share a common part T_c . The labeling information L is integrated in the learning process to enhance the discrimination.	50
4.2	Different CNN structures for RGB-D data. Green, blue, yellow and red boxes indicate convolutional, pooling, fully-connected and softmax layers, respectively.	51
4.3	Illustration of the transformation matrix.	53
4.4	Illustration of the regression coefficient matrix.	53
4.5	Confusion matrix of the category recognition results on RGB-D Object Dataset. The vertical axis shows the true labels and the horizontal axis shows the predicted labels.	62
4.6	Misclassification examples. Misclassifications are due to similar color, texture or geometry shape.	65

4.7	Confusion matrix of the category recognition results on 2D3D Dataset. .	66
4.8	The effect of choosing different R	67
4.9	The effect of choosing different β	68
5.1	Our framework: we first extract proposals from each RGB-D image. Then for all the proposals and the full image, we derive CNN features from different modalities: RGB, HHA and surface normal (SN). For each image, the proposal based CNN features for each modality are encoded by Fisher Vector (FV), and the resulted multi-modal FV features are regarded as the input to our modality and component aware feature fusion. Finally we combine the regression results of the proposal based FV features and the full-image based CNN features to get the final classification result.	72
5.2	Illustration of the component regularization term, where each component is treated as a group and each group is encouraged to have either all zero weights (white squares) or multiple nonzero weights (colorful squares).	76
5.3	Illustration of the modality regularization term, where each modality is treated as a group and each group is encouraged to have sparse nonzero weights (colorful squares) and many zero weights (white squares). . . .	77
5.4	Confusion matrices of ‘FV (L1)’ (left) and ‘FV (Modality+Component+L1)’ (right) on SUNRGBD Dataset. It shows that by adding the modality and component regularization terms, the performance is improved for almost all the classes.	80
5.5	The effect of choosing different λ_2	85
5.6	Confusion matrix of our final results (‘Combine FV and Full’) on NYU Depth Dataset V2.	87

List of Tables

3.1	Class-average accuracy comparison of different methods on the NYU depth dataset V1.	35
3.2	Class-average accuracy results of our method with different settings on the NYU depth dataset V1.	36
3.3	Individual class label accuracy on the NYU depth dataset V1 with only one-layer features. The bold numbers are to indicate the cases that extra depth features hurt the performance. In contrast, the performance is boosted when jointly learning from two modalities for all the categories.	36
3.4	Labeling accuracy on the NYU depth dataset V1 for first-layer, second-layer, both-layer feature learning and feature encoding structure. Second column: the results of our JFLE framework. Third column: the results of our JDFLE framework.	38
3.5	Labeling accuracy of 4-class and 13-Class settings on the NYU depth dataset V2.	42
3.6	Labeling accuracy of 40-class on the NYU depth dataset V2.	42
3.7	Labeling accuracy on the NYU depth dataset V2 for first-layer, second-layer, both-layer feature learning and feature encoding structure using our proposed JFLE and JDFLE frameworks.	42
3.8	The results of different settings of JDFLE.	43
3.9	Individual class labeling accuracy of 4-Class setting on the NYU depth dataset V2.	44
3.10	Individual class labeling accuracy of 13-Class setting on the NYU depth dataset V2.	44

3.11	Individual class labeling accuracy of 40-Class setting on the NYU depth dataset V2.	45
4.1	Comparison of different baselines of using CNNs on RGB-D Object Dataset.	63
4.2	Comparison of our method with the best baseline. Here the CNNs in both methods are pretrained with a subset of ILSVRC2012 dataset. . . .	64
4.3	Comparison with state-of-the-art methods on RGB-D Object Dataset. .	64
4.4	Comparison on 2D3D Dataset.	66
5.1	Comparing the classification results of the proposal based FV features and the full-image based CNN features under different modalities with linear SVM classifier on SUNRGBD Dataset.	82
5.2	Comparison of different baselines of our proposed feature fusion framework on SUNRGBD Dataset.	82
5.3	Comparison with state-of-the-art methods on SUNRGBD Dataset. . . .	83
5.4	Comparing the classification results of the proposal based FV features and the full-image based CNN features under different modalities with linear SVM classifier on NYU Depth Dataset V2.	86
5.5	Comparison of different baselines of our proposed feature fusion framework on NYU Depth Dataset V2.	86
5.6	Comparison with state-of-the-art methods on NYU Depth Dataset V2. We reimplemented the second-order pooling method [5] and show the results as 'O2P'.	87

Chapter 1

Introduction

1.1 Background

Scene understanding is a fundamental and important problem of computer vision, including many tasks such as scene labeling, object detection and recognition, scene classification. It plays an important role in robotics applications including housekeeping and automatic driving, and augmented reality applications which bring enriched interactive experience to people. Scene understanding work started from outdoor scenarios. Compared with outdoor scene understanding, indoor scene understanding could be more challenging since indoor scene could be more complex and cluttered than outdoor. It has been shown in [75] that many scene understanding models that work well for outdoor scenes perform poorly in the indoor domain.

Understanding indoor scene is an easy task for human being even with a single flat image. For example, in Figure 1.1, people can quickly spot wardrobe, bed and nightstand in the scene. In addition to objects, rich contextual information can be inferred, such as: the nightstand is partly occluded by the bed, several props are supported by the nightstand. 3D structure of floor and walls can be quickly known, and the scene can be identified as bedroom.

Figure 1.1: Understanding indoor scene is an easy task for people. Semantic information of objects and the whole scene could be inferred.

For an automatic indoor scene understanding system, the task is challenging due to a number of factors: relatively poor lighting condition, messy object distribution (objects are occluded by each other), and large variance of features for objects and scenes in the same category.

With the wide availability of affordable RGB-D cameras including time-of-flight cameras and structured light cameras [23], it is possible to obtain dense and accurate depth data along with RGB images. For example, Microsoft Kinect [65] is equipped with the infrared camera to generate the depth images by capturing the continuously projected infrared structured light. The depth image generated by Kinect has a resolution of 480×640 and can be automatically calibrated with the RGB image. The synchronized depth information brings rich 3D geometry information. Combination of color and depth information provides a great chance to significantly boost the performance on

indoor scene understanding.

Various approaches have been proposed for RGB-D scene understanding. In general, they can be divided into two categories from the perspective of feature generation: methods with hand-crafted features and methods with learned features. Hand-crafted features such as *Scale-invariant feature transform* (SIFT) [63] and *Local binary patterns* (LBP) [72] are used to describe color and texture information of color images and 3D geometry information of depth images. The problem with hand-crafted features is that they cannot be readily extended to different datasets or other modalities, since they are often manually tuned for the conditions encountered in the studied datasets. In addition, hand-crafted features can only capture a subset of the cues that are useful for recognition. To minimize the need of hand-crafted features, many methods have been proposed to learn features from raw data.

This thesis investigates feature learning for scene understanding using RGB-D input with different focuses. In particular, we explore three different tasks: scene labeling, object recognition and scene classification.

1.2 Objective and Research Scope

1.2.1 Scene Labeling

Scene labeling is an integral part of scene understanding and involves densely assigning a category label to each pixel in an image. Most previous scene labeling works dealt with outdoor scenarios [22, 24, 28, 57, 66, 89]. For indoor scenes, with the help of low-cost RGB-D cameras, noticeable improvements in accuracy and robustness of the scene labeling task have been achieved.

Hand-crafted features were used in several previous works on RGB-D scene labeling. These include the use of SIFT [84], LBP [72], *Histogram of oriented gradients* (HOG) [16], texon [45], spin image [84], class-specific location [45], planarity [12], height above the ground, 3D shape contexts, size features [30], and some sophisticated features such as *kernel descriptors* (KDES) [76].

To reduce the dependency of hand-crafted features, several works have applied feature learning for RGB-D indoor scene labeling. In [73], pixels of patches are encoded with example patches selected from input data. Multi-scale *convolutional neural networks* (CNN) is used in work [15] for RGB-D feature learning. Both methods obtained limited performance for indoor scene labeling.

Recently, unsupervised feature learning has achieved great success in many applications including object recognition [52] and action recognition [53], where features learned via unsupervised method are used as local features instead of previous hand-crafted features. It is shown that better performance can be achieved by further applying feature encoding over the learned features to build “bag of words” type of features in [53, 97]. However, they perform feature learning and feature encoding separately, which might cause inconsistency between the two components since feature learning is not optimized for feature encoding and feature encoding is also not optimized for feature learning.

Such research gap between feature learning and encoding motives us to design a joint framework in Chapter 3 for RGB-D scene labeling.

1.2.2 Object Recognition

The target of object recognition is to classify an RGB-D image containing one object in it. Recognizing commonplace objects of indoor scenes is a challenging task as objects in real-world indoor rooms have large variation both in color and geometry.

Similar with scene labeling task, quite a few methods have been proposed to extract hand-crafted features to describe the object images [48, 84]. In addition, to relief the dependency on hand-crafted features, several recent methods have been proposed for feature learning from raw data directly. There is a substantial leap in object recognition [49] performance using *convolutional neural networks* (CNN) trained on large-scale datasets such as ImageNet [19]. For RGB-D object recognition, representative methods include convolutional-recursive deep learning [88], hierarchical matching pursuit [9, 10], convolutional k-means descriptors [7], hierarchical sparse coding [107] and local coordinate coding [98].

However, the relationship between different modalities have not been thoroughly investigated in these feature-learning methods for RGB-D object recognition. Most of the methods either learn features from color and depth modalities separately, or simply treat RGB-D as undifferentiated four-channel data. The major shortcoming for the separate learning of these methods is that the relation between the two modalities is ignored and the feature learning of one modality is not adjusted by the other modality. The major shortcoming for a simple four-channel learning is that the combination may not be physically meaningful and may not capitalize on different characteristics of the modalities.

Thus, in Chapter 4 of this thesis, we explore the relation of multiple modalities based on deep learning structure for RGB-D object recognition.

1.2.3 Scene Classification

Classifying scene is a difficult problem, especially for indoor environment, due to the large intra-class variation with vast differences in spatial layouts within each scene class. To investigate the utility of CNN in scene classification tasks, a scene-centric dataset known as *Places* [114] has been introduced. Although there was reported performance

improvement using a scene-centric CNN, it became obvious that global CNN features extracted from full images were too spatially rigid to be optimal for scene classification.

Several methods [26, 106, 116] have been proposed for classifying RGB scene images using local instead of global information. They share a similar pipeline: CNN features were densely extracted at different locations and scales of an image, encoded as a combined feature representation (e.g. via *Fisher vectors* (FV) [74, 79]) and then classified using *support vector machines* (SVM). Results show that the local features are competitive when compared to full-image based CNN features and provide important complementary information. However, only a small subset of local features are likely to be discriminative in a scene classification task. In many existing works, a task-independent feature representation is used, such as a comprehensive *Gaussian mixture model* (GMM) which models all features for encoding Fisher vectors; this tends to result in overfitting when training regressors.

There are also a few methods proposed for scene classification on RGB-D images [5, 30, 60, 90]. Most of these directly concatenate features from color and depth modalities together prior to classification. Such a direct combination does not adequately exploit the relationship between the different modalities of color and depth.

Thus, in Chapter 5, we propose a framework to effectively fuse Fisher Vector features from different modalities.

1.3 Summary of Contributions

The thesis has made several contributions to the RGB-D scene understanding related research with different focuses.

- For scene labeling task, we propose an unsupervised *joint feature learning and encoding* (JFLE) framework for RGB-D scene labeling. The main novelty of our learning framework lies in the joint optimization of feature learning and feature encoding in a coherent way which significantly boosts the performance. By stacking basic learning structure, higher-level features are derived and combined with lower-level features for better representing RGB-D data. Moreover, to explore the nonlinear intrinsic characteristic of data, we further propose a more general *joint deep feature learning and encoding* (JDFLE) framework that introduces nonlinear mapping into JFLE. Experimental results show that our approaches achieve competitive performance, compared with state-of-the-art methods, while our methods do not need complex feature handcrafting and feature combination and can be easily applied to other datasets.
- For object recognition task, motivated by the intuition that different modalities should contain not only some modal-specific patterns but also some common patterns, we propose a multi-modal feature learning framework for RGB-D object recognition. We first construct deep CNN layers for color and depth separately, and then connect them with our carefully designed multi-modal layers, which fuse color and depth information by enforcing a common part to be shared by features of different modalities. In this way, we obtain features reflecting shared properties as well as modal-specific properties in different modalities. Supervised information is further integrated into the framework to extract more discriminative features.
- For scene classification task, we investigate a framework allowing greater spatial flexibility, in which the Fisher vector encoded distribution of local CNN features, obtained from a multitude of region proposals per image, is considered. The CN-

N features are computed from an augmented pixel-wise representation comprising multiple modalities of RGB, *Horizontal disparity*, *Height above ground*, and *Angle between the local surface normal and direction of inferred gravity* (HHA) and surface normals, as extracted from RGB-D data. More significantly, we make two postulates: (1) component sparsity — that only a small variety of region proposals and their corresponding FV GMM components contribute to scene discriminability, and (2) modal non-sparsity — within these discriminative components, all modalities have important contribution. In our framework, these are implemented through regularization terms applying group lasso to GMM components and exclusive group lasso across modalities. By learning and combining regressors for both proposal-based FV features and global CNN features, we are able to achieve encouraging results highly competitive to the state-of-the-art algorithms.

1.4 Thesis Organization

The remainder of this report is organized into the following parts: Chapter 2 reviews the related work of scene understanding and some related techniques of feature learning. In Chapter 3 we introduce our unsupervised joint feature learning and encoding method for RGB-D scene labeling. In Chapter 4 we present the multi-modal sharable and specific feature learning method for object recognition. In Chapter 5 we propose a modality and component aware feature fusion method for scene classification. Finally, we conclude our work and provide some future directions in Chapter 6.

Chapter 2

Literature Review

In this chapter, we mainly review the existing research works related to RGB-D scene understanding. In particular, methods for scene labeling, object recognition and scene classification are reviewed. Also, we review related literature in feature learning.

2.1 Scene Labeling

Early work on scene labeling focused on outdoor color imagery, and typically used *Conditional Random Field* (CRF) or *Markov Random Field* (MRF). The nodes of the graphical models were pixels [33,83], superpixels [24,66] or a hierarchy of regions [57]. Local interactions between nodes were captured by pairwise potentials, while unary potentials were used to represent image observations, via features such as SIFT [63] and HOG [16]. An alternative inference framework was presented in [89], in which a very efficient *recursive neural network* (RNN) was used to greedily merge neighboring superpixels according to a learned scoring function. In a departure from the earlier approaches involving hand-crafted feature extraction, Grangier *et al.* [28] used convolutional networks for scene labeling. Farabet *et al.* [22] later adopted multiscale convolutional networks to automatically learn low and high-level texture and shape features from raw

pixels, and further proposed the “purity” of class distributions as an optimization goal, in order to maximize the likelihood that each segment contains only one object. They achieved state-of-the-art performance on the commonly used Stanford Background [27] and SIFT Flow datasets [61]. Indoor scene labeling is a harder problem, but it has become more accessible recently with the advent of affordable RGB-D cameras such as Kinect.

Hand-crafted feature based methods for RGB-D scene labeling: Silberman and Fergus [85] released a large-scale RGB-D dataset containing 7 scene types and 13 semantic labels. They employed RGB-D SIFT and 3D location priors as features and used MRFs to ensure contextual consistency. Koppula *et al.* [48] achieved high accuracy on semantic labeling of point clouds via a mixed integer optimization method. They however require the extraction of richer geometry features from 3D+RGB point clouds rather than the more limited height field from a single RGB-D image, and also depend on a computationally intensive optimization process with long running time. In an extension to Silberman and Fergus’s work, Ren *et al.* [76] evaluated six kernel descriptors and eventually chose four of them. Besides, more comprehensive geometry features of superpixels were added to further boost the performance. With the additional features, we achieve a performance improvement to the previous state-of-the-art method [76] on the NYU depth dataset V1. Cadena and Kosecka [12] proposed various new features including entropy for associating superpixel boundaries to vanishing points, and neighborhood planarity. Gupta *et al.* [30] proposed an object boundary detection method which naturally combines color and depth information. With better bottom-up segmentation, they extract generic and class-specific features to encode superpixels. Muller *et al.* [68] proposed to fuse color, depth information with a random forest approach. They built a CRF model not only based on spatial relations in 2D but also geometry relations in 3D. Khan *et al.* [45] proposed a CRF framework with long range interaction based

on geometry features as higher order potentials besides the unary and pairwise potentials. Hermans *et al.* [34] proposed a CRF-based scene labeling method and created a consistent 3D semantic reconstruction of indoor scenes based on 2D results. All these methods mentioned above require manual fine-tuning in feature design and also in the combination of different features.

Learned feature based methods for RGB-D scene labeling: To reduce the dependency on hand-crafted features, several feature learning methods have been proposed to learn features for RGB-D scene labeling. Pei *et al.* [73] learned features by projecting raw pixels of patches onto selected example patches. Such an encoding method may not be powerful enough since the input raw pixel values are usually redundant and noisy. Lai *et al.* [50] proposed an unsupervised method for labeling point clouds of tabletop objects with RGB-D videos as the input. They presented a hierarchical sparse coding technique for learning features from 3D point cloud data. Recently, convolutional neural network has consistently achieved superior performance in computer vision tasks. Couprie *et al.* [15] applied the *convolutional neural network* (CNN) method of Farabet *et al.* [22] to indoor RGB-D scene labeling. The depth data was treated as an additional channel besides RGB, and a multiscale convolutional network was used to ensure the features capture a larger spatial context. Although this method was demonstrated to be effective for outdoor scenes, the performance on RGB-D indoor scenes is much less satisfactory. Eigen and Fergus [21] proposed to use an end-to-end CNN structure to predict depth, surface normals and semantic labels. Instead of extracting CNN features from superpixels, they directly predicted pixel-level labels. To our knowledge, their method is current state of the art. These CNN-based methods are supervised, while our method in Chapter 3 focuses on unsupervised learning.

2.2 Object Recognition

Hand-crafted feature based methods for RGB-D object recognition: Many methods focusing on the design of hand-crafted features have been proposed for RGB-D object recognition. For example, Lai *et al.* [51] used hand-crafted features including spin images [42] and SIFT descriptors [64] for depth images, while textons [59], color histograms [1] and SIFT descriptors were used for color images. Lai *et al.* also utilized the bag-of-words-based *Efficient Match Kernel* (EMK) to encode local hand-crafted features, and derived an image-level representation via integrating EMK features in different spatial parts. Using the encoded features, they evaluated the recognition performance of different classifiers: a linear support vector machine, a Gaussian kernel support vector machine and a random forest classifier. Bo *et al.* [8] developed a set of kernel features for depth images that model sizes, 3D shapes, and depth edges to further improve recognition performance.

Learned feature based methods for RGB-D object recognition: To minimize the need of hand-crafted features, several recent methods have been proposed for unsupervised learning of features from raw data directly, to be used in RGB-D object recognition. In particular, Bo *et al.* [10] proposed to use a *Hierarchical Matching Pursuit* (HMP) method [8] based on sparse codes derived not only from RGB-D images but also gray-scale intensities and surface normals, computed via K-SVD [2]. These features captured high-level information from local patches. Blum *et al.* [7] described a feature learning approach which learns dictionaries from RGB-D data based on K-means clustering of local-patch features, where the image patches are extracted around the interest points detected by SURF features [6]. Socher *et al.* [88] outlined a framework which integrated *Convolutional Neural Networks* (CNN) and *Recursive Neural Networks* (RNN) to learn features from color and depth separately, where the single-layer CNN is pre-

trained in an unsupervised manner to produce lower-level features while the RNN learns higher-level features.

For existing methods that learn features from color and depth modalities separately [88] and concatenate them prior to classification, the major shortcoming is that the relation between the two modalities is ignored and the feature learning stage of one modality would not be effected by other modalities, and thus the complementary nature of different modalities cannot be fully exploited. For other methods that simply combine the modalities from the outset [7, 9, 10] and adopt a conventional approach to learn features, the major shortcoming is that the combination is not physically meaningful.

2.3 Scene Classification

Object recognition performance has recently been boosted through the use of well designed CNN techniques [49] in conjunction with extensive labeled data. To adapt the current CNN techniques for scene classification, Zhou *et al.* [114] introduced a large scene-centric dataset called Places and showed significant performance improvement on scene classification using a CNN trained on this dataset, as compared to directly applying the CNN pretrained on the object-centric dataset ImageNet [19]. Although a scene-centric dataset more appropriately captures the richness and diversity of scene imagery, the typical way of extracting global CNN features from full images may not adequately handle the geometric variability of complex indoor scenes.

Several methods have been proposed to leverage local CNN features to enhance discriminative capability. Gong *et al.* [26] proposed densely extracting multi-scale CNN activations, aggregating the activations of each scale via *vector of locally aggregated descriptors* (VLAD) [39], and concatenating the multi-scale VLAD features together

as the final feature representation. Yoo *et al.* [106] presented a similar framework, except they used *Fisher Vectors* (FV) as the encoding method. In another work [116], Zuo *et al.* showed the importance of the complementary information provided by local features, where they derived local features by learning a discriminative and shareable feature transformation filter bank for local image patches. Among all these methods, few of them take direct care to exclude non-discriminative local features that can lead to overfitting.

There are also several other works that are not developed for scene classification, but related to our method. In particular, Yang *et al.* [102] approached the multi-label image classification problem through multi-view learning, where they derived a feature view by extracting CNN features from object proposals followed by the FV encoding, and constructed a label view using strong labels. Zhang *et al.* [113] dealt with fine-grained image categorization, where they proposed to use feature selection to remove noisy features in FV. Their feature selection is based on the relevance of individual features to class labels, which are calculated independently in different feature dimensions.

On the topic of RGB-D scene classification, Gupta *et al.* [29, 30] described a method to detect contours in RGB-D images and use them for semantic segmentation, further treating the quantized semantic segmentation output as local features for scene classification. Banica *et al.* [5] proposed to apply second-order pooling [13] of hand-crafted features mainly for semantic segmentation as well as on scene classification. Song *et al.* [90] introduced a large scale RGB-D dataset called the SUNRGBD Dataset with ground truth and baselines for different scene understanding tasks. For scene classification, they directly used pre-trained CNN in [114] to extract CNN features from RGB and HHA. Liao *et al.* [60] proposed to include a regularization on semantic segmentation to improve scene classification performance, where their cost function to train CNN contains both the loss of scene classification and the loss of semantic segmentation.

2.4 Feature Learning

Deep Learning: A number of deep learning methods have been proposed in computer vision in recent years, which aim to learn invariant and hierarchical feature representations for different applications. Representative deep learning methods include deep belief networks [35], convolutional deep belief networks [56], deep Boltzmann machines [78], and stacked denoising autoencoders [94]. Most of them have demonstrated promising performance in visual analysis.

Among these deep learning methods, deep convolutional neural networks have produced excellent performance in a number of vision tasks. In 1998, LeCun *et al.* [54] proposed a convolutional neural network structure to deal with handwritten digit recognition problem. In 2012, another representative work was proposed by Krizhevsky *et al.* in [49]. They achieved superior image classification performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [18, 19] with a large CNN using 1.2 million labeled images. Techniques such as dropout, data augmentation and engineering trick of adjusting step size help to make CNN more trainable. Since then, tremendous interests have been attracted to this deep learning field. CNNs provide consistently high performance on various vision tasks such as scene labeling [22], object detection [25], video classification [44], house number digit classification [81], image super-resolution [20], image quality assessment [43], action recognition [40], face verification [91], pedestrian detection [110], and pose estimation [93]. In addition, deep convolutional neural networks start to show advancement on text understanding from word-level [67] or character-level [112] inputs. There are also a few methods [109] [87] which visualize the intermediate features of CNNs to help us understand the insight of what CNNs learn and adjust the network design accordingly.

In Chapter 4, we focus on RGB-D object recognition by integrating multi-modal feature learning with CNN. In Chapter 5, we use CNN to extract local and global features of scenes. There are also some approaches that use deep learning for other tasks on RGB-D data. For example, Lenz *et al.* [58] proposed a deep learning method for robotic grasps detection. They used stacked auto-encoder structures and derived multi-modal features by encouraging each dimension of the learned features using information from only a subset of the modalities. Hu *et al.* [36] focus on the RGB-D activity recognition task by fusing multiple different hand-crafted features.

Multi-modal methods: We wish to point out that our work on RGB-D object recognition in Chapter 4 is inspired by the multiview learning method proposed by Liu *et al.* in [62], which explores the consistency and complementary properties in multiview data. However, their method is based on nonnegative matrix factorization and focuses on semi-supervised learning, which cannot handle the data unseen in the training stage. In contrast, our multi-modal feature learning framework is based on matrix transformation which extracts both shared common patterns and modal-specific properties of different modalities, and is integrated with CNN based supervised deep learning for RGB-D object recognition. Jia *et al.* [41] and Klami *et al.* [46] proposed methods which share similar idea with our method. Jia *et al.* [41] proposed to factorize the information into shared and private parts using sparse coding and structure sparsity for human pose estimation. Klami *et al.* [46] proposed a dependency-seeking clustering algorithms with variational Bayes. In addition, the method of Daumé [17] shares a similar idea with our method but focuses on transfer learning. Daumé proposed to augment data by containing general and specific versions of features. With the augmented data from both domains, their method trains a classifier which could classify data from the target domain, while leveraging information from both the source and target domains. In another

recent work [111], Zhang *et al.* [111] proposed a new *discriminative canonical correlation analysis* (DCCA), which regards color as the main modality and depth as the auxiliary information for an object recognition task. They considered a transfer learning setting where color and depth images are used in training while only color images are used in testing. Their target task is different from ours.

Structure Sparsity: Our proposed method for RGB-D scene classification in Chapter 5 utilizes the methods of structure sparsity. Structure sparsity is an extension of the standard sparsity concept, which aims to facilitate arbitrary structures on the feature set [37]. The effectiveness of structure sparsity for feature learning has been widely proven in different applications such as face recognition [99], web page recognition [115], image super resolution [103], action recognition [105], and object recognition [69].

Here we discuss several representative pieces of research that are relevant to our method. In particular, Tibshirani [92] proposed the idea of “lasso” (least absolute shrinkage and selection operator) which minimizes the squared errors with an l_1 -norm regularization term. It essentially shrinks some coefficients and sets others to 0. The relationship between the loss function and the regularization term is analyzed in [92].

Yuan and Lin [108] further lasso for variable selection with predefined groups, which is usually called “group lasso”. Their key assumption is that if a few features in a group are important, then the whole group is regarded as important. For tasks benefiting from the selection of important groups, their method improves the performance of the traditional lasso. Zhou *et al.* [115] further developed a new form of regularization called “exclusive lasso”, where they focused on multi-task feature selection. Their assumption is that features that are important for one category become less likely to be important for other categories, and thus their idea is to introduce the competition among different tasks for the same feature. Kong *et al.* [47] shared a similar idea with Zhou *et al.* [115], but they

focused on feature selection with multi-group of features. They proposed “exclusive group lasso” to encourage features in different groups to co-exist, which is different from group lasso that enforces inter-group sparsity. Combining with the traditional lasso, exclusive group lasso demonstrates its effectiveness on the spoken letter classification task [115]. In Chapter 5, we combine both group lasso and exclusive group lasso in our feature fusion framework to solve the scene classification problem.

Chapter 3

Unsupervised Joint Feature Learning and Encoding for RGB-D Scene Labeling

The approach proposed in this chapter attempts to learn visual patterns from RGB-D input via an unsupervised learning framework. At the heart of our unsupervised learning algorithm, we perform feature learning and feature encoding jointly in a two-layer stacked structure, called *joint feature learning and encoding* framework (JFLE), which is very different from the conventional way of feature learning followed by feature encoding (e.g. BOW, sparse coding), i.e. two separate processes. Moreover, to explore the nonlinear intrinsic characteristics of data, we further extend the JFLE framework to a more general framework called *joint deep feature learning and encoding* (JDFLE), which uses a deep model with stacked nonlinear layers to model the input data. Fig. 3.1 illustrates the proposed overall framework. The input to the learning structure (either JFLE or JDFLE) is a set of patches densely sampled from RGB-D images, and the learning output is the set of corresponding path features, which are then combined to generate superpixel features. Finally, linear SVMs are trained to map superpixel features to scene labels.

Figure 3.1: Our framework for RGB-D indoor scene labeling. Our method learns features from raw RGB-D input with two-layer stacking structure. Features of the two layers are concatenated to train linear SVMs over superpixels for labeling task.

The contributions of this chapter are threefold. First, we propose an unsupervised joint feature learning and feature encoding framework (JFLE) that makes feature learning and feature encoding help each other in a coherent manner. Second, we further develop a more general JDFLE framework which replaces the linear mapping in JFLE by multiple nonlinear sub-layers. Third, we apply our joint frameworks on RGB-D scene labeling. We conduct extensive experiments to demonstrate the effectiveness of the proposed joint frameworks.

3.1 Joint feature learning and encoding

In this section, we describe our proposed basic feature learning framework, called *joint feature learning and encoding* (JFLE), which performs joint feature learning and encod-

Figure 3.2: Detailed illustration of feature extraction of the first layer: $s \times s$ color and depth patches are flattened into vectors X and Y . Z is the input vector obtained by concatenating X and Y . With learned filter matrix W and dictionary U , the sparse encoding coefficients V can be derived, which represents the feature of a $s \times s$ patch. By concatenating the features of $(S-s+1)^2$ $s \times s$ small patches, we get the feature of a $S \times S$ big patch.

ing in one optimization framework.

3.1.1 Single-layer feature learning structure

Our approach is based on the unsupervised feature learning algorithm [52], which is to minimize the following objective function

$$\min_W \|W^T WZ - Z\|_2^2 + \lambda_1 g(WZ) \quad (3.1)$$

where Z is a set of d -dimensional raw input data vectors, i.e. $Z = [z_1, \dots, z_m] \in \mathbb{R}^{d \times m}$, $W \in \mathbb{R}^{d' \times d}$ is the transform matrix which projects Z into a d' -dimensional feature space, g is the smooth L_1 penalty function [52], and λ_1 is a tradeoff factor. Eq. (3.1) essentially seeks the transformation matrix W that can minimize the reconstruction error (first term) and the smooth L_1 penalty on learned features WZ (second term). The transform matrix $W \in \mathbb{R}^{d' \times d}$ is often chosen to be overcomplete, i.e. $d' > d$, for better performance, as demonstrated in the study [14]. Note that Z has gone through the whitening preprocess, i.e. the input data vectors are linearly transformed to have zero mean and identity covariance [52]. Such unsupervised feature learning method has been proven to be successful in the application of object recognition [52].

Previous methods [53, 97] show that better performance can be achieved by further applying feature encoding over the learned features to build “bag of words” type of features. However, they perform feature learning and feature encoding separately, which might cause inconsistency between the two components since feature learning is not optimized for feature encoding and feature encoding is also not optimized for feature learning.

Therefore, motivated by the above observation, in this chapter we propose to perform feature learning and feature encoding in a joint framework with the following objective

function:

$$\begin{aligned} \min_{W, V, U} \quad & \|W^T WZ - Z\|_2^2 + \lambda_1 g(WZ) \\ & + \lambda_2 \|WZ - UV\|_2^2 + \lambda_3 |V|_1 \end{aligned} \quad (3.2)$$

$$\text{subject to} \quad \|u_k\|_2 \leq 1, k = 1, 2, \dots, K.$$

where $U = [u_1, \dots, u_K] \in \mathbb{R}^{d' \times K}$ represents the dictionary which has K bases, and V denotes the feature encoding coefficients. Compared with Eq. (3.1), the newly added two terms in Eq. (3.2) aim to find sparse feature representation for the learned feature WZ . At the same time, there is a L2-norm constraint for u_k to avoid trivial solutions which just scale down V and scale up U . By jointly learning W , V and U in Eq. (3.2), we integrate feature learning and feature encoding into a coherent framework, where the two parts are optimized to help each other. With the optimized W , transformed data WZ could be encoded by more descriptive dictionary U and the final features V become more efficient.

In particular, for RGB-D scene labeling we learn multi-modality features. Instead of learning W for color and depth information separately, we consider different modalities jointly and their relationship is implicitly reasoned. Specifically, let $X = [x_1, \dots, x_m] \in \mathbb{R}^{d_1 \times m}$ denote the input RGB vectors, and $Y = [y_1, \dots, y_m] \in \mathbb{R}^{d_2 \times m}$ denote the input depth vectors. Then, Z in Eq. (3.1) is simply formed by cascading color and depth information as $Z = [X; Y] \in \mathbb{R}^{d \times m}$ ($d = d_1 + d_2$).

3.1.2 Optimization Process

In the proposed unsupervised feature learning Eq. (3.2), we need to optimize W , U and V together. We solve this problem by updating three variables iteratively. W , U and V are initialized randomly. Given a training data matrix Z , we first fix U and V , the cost

Figure 3.3: Left: the unsupervised learning structure of the first layer. Right: the second layer structure. F_1 is the first-layer feature. F_2 is the second-layer feature.

Algorithm 1: Optimization process.

Input: Raw data from multiple modalities: Z

Output: Transformation matrix W , Dictionary U , Sparse encoding V

Step 1: Initialization.

W , U and V are randomly initialized;

Step 2: Iteratively optimize over W , U and V . while $iter \leq max_iter$ do

 Fix U and V in Eq. (3.2):

 Can be solved by unconstrained optimizer L-BFGS and update W

 Fix W and U in Eq. (3.2):

 A linear regression problem Eq. (3.3) over V with L1 norm regularization on the coefficients.

 Optimized by feature-sign search algorithm and update V

 Fix W and V in Eq. (3.2):

 A least square problem Eq. (3.4) with quadratic constraints over U

 Optimized by Lagrange dual and update U

end

function can then be minimized by using the unconstrained optimizer (e.g. *Limited-memory Broyden-Fletcher-Goldfarb-Shanno* (L-BFGS) method [80], *Conjugate Gradient* (CG) method [80]) to update W . When fixing W and U , similar to the sparse coding work [104], Eq. (3.2) becomes a linear regression problem with regularization on the coefficients, which can be solved efficiently by optimization over each coefficient v_m with the feature-sign search algorithm [55]. At last, when W and V are fixed, it becomes a least square problem with quadratic constraints, which can be easily solved. The optimization process is shown in Algorithm 1.

$$\min_V \lambda_2 \|WZ - UV\|_2^2 + \lambda_3 |V|_1 \quad (3.3)$$

$$\begin{aligned} \min_U \quad & \lambda_2 \|WZ - UV\|_2^2 \\ \text{subject to} \quad & \|u_k\|_2 \leq 1, k = 1, 2, \dots, K. \end{aligned} \tag{3.4}$$

3.1.3 Hierarchical feature learning

What we present in section 3.1.1 is just a one-layer feature learning structure. Considering that there exists multi-level information in visual data such as intensity, edge, object, etc [56], it is often preferred to learn hierarchical features so as to describe low-level and high-level properties simultaneously. In our case, we stack the single-layer feature learning structure to capture the higher-level features. Particularly, we first learn the low-level features using the single-layer structure. Then, the output of the low-level structure V is treated as the input for the higher level. Considering the output of the first-layer learning structure is of high dimension, PCA is used to reduce its dimension so that the same structure can be reused for the high-layer feature learning. We use PCA for efficiency, otherwise the dimension of the output of the first layer will be too high to process in the second layer. In the stacked structure, the input Z of higher level would contain lower-level features from the two modalities produced by the lower-level feature learning.

3.1.4 Application on RGB-D scene labeling

For the RGB-D scene labeling application, when the input data has large size, the learning process becomes less efficient. To address this, we make use of small patch features to represent big patches. Our main framework for RGB-D labeling is as follows. We first run our unsupervised learning on randomly sampled small patches ($s \times s$) to learn the optimal transform matrix W and the dictionary U . Then, for each densely sampled big patch ($S \times S, S > s$), with the obtained W and U we derive the feature vector V for its

overlapped $s \times s$ small patches. Features of $S \times S$ patches are then obtained by concatenating all its overlapped $s \times s$ patches' features together. Finally, superpixel technique is incorporated to ensure that pixels in the same superpixel take the same label.

Fig. 3.2 shows the detailed first-layer feature extraction process. In particular, we extract input raw data from two different modalities (color and depth). We convert the color image to grayscale. At the beginning, m $s \times s$ RGB-D small image patches are randomly sampled. For each $s \times s$ small patch, X is s^2 -d raw color data by flattening the patch into a vector. The same goes for raw depth data Y . Concatenating them together, we have Z , $2s^2$ -d data. For each $S \times S$ big patch, there are $(S - s + 1)^2$ $s \times s$ small patches. After the unsupervised feature learning process, a small patch is then represented by a sparse vector V (K -dimensional) computed from W and U . Concatenating the features of $(S - s + 1)^2$ small patches together, we obtain the features of a big patch. To avoid over-fitting caused by the high dimensionality of the big patch features, we use max-pooling to reduce the dimensionality. We conducted max pooling among $(S - s + 1)^2$ small patches.

To capture higher-level features, we stack two single-layer structure together. Fig. 3.3 shows the two-layer feature learning structure, where the output features of the first layer are used as the input for the second layer. Specifically, the first-layer output feature vectors are further processed through dimension reduction by PCA so that the vectors could be resized to $S \times S$ data patches. Same as the first layer, $s \times s$ small patches in these $S \times S$ big patches are sampled as training data of the second layer. After the learning process of the second layer, these $S \times S$ patches are represented by the concatenated features of their $s \times s$ patches. At last, the features from the two different layers are concatenated together as the final representation of the raw patches.

In our patch size setting, we set S as 10 and s as 7 for both layers. The input data is normalized between the two modalities. We choose the dictionary size K as 1024. With

learned W and U , the output of the first layer is 1024-d V . After PCA transformation, it is rescaled as 100-d data. The 100-d data is then resized to 10×10 patches, where the overlapping 7×7 patches are the training input for the second layer. By concatenating 16 1024-d features, we get a 16384-d feature vector for a 10×10 patch. Then, max-pooling is used to reduce the dimension to 1024-d for one layer. Concatenating the features of the two layers, we finally obtain a 2048-d feature for each 10×10 patch.

After feature learning process, scene labeling is done using the learned patch features. Considering that predicting the pixel-wise labeling independently could be noisy and pixels with the same color in local regions should take the same label, we oversegment RGB-D images using *globalized Probability of boundary* (gPb) hierarchical segmentation method [3], where we follow the adaptation to RGB-D images proposed by Ren *et al.* [76] to linearly combine the *Ultrametric Contour Maps* (UCM) results of RGB and depth. UCM is the output of the gPb oversegmentation method. The 10×10 patches are obtained by densely sampling over a grid with a unit distance of eight pixels. Finally, each superpixel is represented by averaging the features of all the patches whose centers are located in the region.

3.2 Joint deep feature learning and encoding

By simultaneously performing feature learning and feature encoding, JFLE can obtain more efficient transformation matrix W and more descriptive dictionary U , compared with separate learning and encoding. Note that JFLE is under the assumption that the transformed data are vectors in an Euclidean space, meaning that each vector can be represented by the linear combination of a small number of vectors in dictionary. However, the original RGB-D data might need more complex description of the nonlinear

manifold, as indoor objects usually have large variation in appearance. The linear assumption might be inappropriate as it ignores the important intrinsic characteristic of data. In this section, we introduce nonlinear mapping as an extension to our joint feature learning and encoding method. Instead of using transformation matrix to generate features, we use nonlinear mapping to map raw data to a more descriptive feature space. There have been works about nonlinear mapping. For example, Xie *et al.* [101] proposed to optimize the dictionary and sparse coding on a Riemannian manifold instead of Euclidean space. Shirazi *et al.* [82] presented a visual tracking system based on distance computed via non-Euclidean geometry of Grassmann manifolds. Harandi *et al.* [32] proposed to conduct sparse coding and dictionary learning for the Riemannian structure of Symmetric Positive Definite matrices.

We introduce stacked nonlinear layers to model a flexible nonlinear mapping. By doing so, we extend our framework of joint feature learning and feature encoding to a more general nonlinear model called *joint deep feature learning and encoding* (JDFLE), which incorporates intrinsic characteristic of the input data.

3.2.1 Formulation

Fig. 3.4 illustrates one single-layer structure of JDFLE. Instead of just using one projection matrix to model the mapping, we construct a neural network to learn features of objects. Raw RGB-D data are passed through multiple nonlinear transformation sub-layers. For example, passing the input data Z through one nonlinear sub-layer results in $A^{(1)}(Z) = t(W_1 Z + b_1)$, where W_1 denotes the projection matrix to be learned, b_1 denotes the bias, and t is a nonlinear activation function. In this way, data are modeled in a more complex manifold, the discriminative nature could be captured better, and the sparse coding process would not loss the important information. Similar to the linear framework of JFLE, the nonlinear framework of JDFLE also performs feature learning and

Figure 3.4: Illustration of one single-layer structure in the proposed nonlinear framework. Given raw data of an RGB-D patch, we map it with several nonlinear sub-layers (here we have 2 sub-layers): each sub-layer contains linear projection matrix W , bias b and a nonlinear activation function. With the constraints of feature encoding (sparse coding), and the reconstruction constraints for each sub-layer (reconstruction and regularization of bias b), better dictionary and feature mapping could be learned.

feature encoding jointly, while the key difference is that the feature learning of JDFLE maps the original data to a nonlinear feature space which could be better sparse coded.

The notation of the output of one nonlinear sub-layer can be expressed as

$$A^{(m)}(Z) = t(W_m A^{(m-1)}(Z) + b_m) \quad (3.5)$$

with $A^{(0)}(Z) = Z$. With the nonlinear features, we perform feature learning and feature encoding jointly by adding feature encoding constraints on the top sub-layer output, and at the same time apply the reconstruction constraints and the bias regularization constraints on each sub-layer. Specifically, we minimize the following objective function:

$$\begin{aligned} \min_{W, b, V, U} F &= F_1 + F_2 + F_3 \\ &= \left(\|A^{(M)}(Z) - UV\|_2^2 + \xi_1 \|V\|_1 \right) \\ &\quad + \lambda \sum_{m=1}^M \left(\|W_m^T W_m A^{(m-1)}(Z) - A^{(m-1)}(Z)\|_2^2 \right. \\ &\quad \left. + \xi_2 g(W_m A^{(m-1)}(Z)) \right) + \xi_3 \sum_{m=1}^M \|b_m\|_2^2 \end{aligned} \quad (3.6)$$

$$\text{subject to } \|u_k\|_2 \leq 1, k = 1, 2, \dots, K.$$

where F_1 denotes the constraints of feature encoding (i.e. the sparse coding criteria for the nonlinearly-mapped data), F_2 denotes the reconstruction constraints for each sub-layer (i.e. for sub-layer m , data can be reconstructed using the transposed projection matrix W_m^T), F_3 denotes the regularization constraints for the bias b_m for each sub-layer, M is the total number of sub-layers, and ξ_1, ξ_2, ξ_3 and λ are tradeoff parameters, which are positive.

3.2.2 Optimization

To solve the optimization problem in (3.6), we iteratively optimize $\{W, b\}$, U and V . Particularly, when fixing W and b to optimize U or V , the operation is the same as that

in JFLE (see Algorithm 1). When fixing U and V , we optimize W and b of each sub-layer by gradient descent process [4]. The gradients of the objective function F with respect to W_m, b_m for $m = 1, 2, \dots, M$ can be derived as

$$\frac{\partial F}{\partial W_m} = \Phi_1^{(m)} A^{(m-1)T} + \Phi_2^{(m)} \quad (3.7)$$

$$\frac{\partial F}{\partial b_m} = \Phi_1^{(m)} + \xi_3 b_m \quad (3.8)$$

with

$$\Phi_1^{(M)} = 2(A^{(M)} - UV) \otimes t'(H^{(M)}) \quad (3.9)$$

$$\Phi_2^{(M)} = C^{(M)} \quad (3.10)$$

$$\Phi_1^{(m)} = W_{m+1}^T \Phi_1^{(m+1)} \otimes t'(H^{(m)}) \quad (3.11)$$

$$\Phi_2^{(m)} = C^{(m)} + \left(\sum_{k=m+1}^M R^{(k)'} A^{(m)} \right) \otimes t'(H^{(m)}) A^{(m-1)T} \quad (3.12)$$

where $\otimes$ denotes the element-wise multiplication operation, and $C^{(m)}$, $R^{(m)}$ and $H^{(m)}$ are defined as

$$\begin{aligned} C^{(m)} = & 2W_m A^{(m-1)} (W_m^T W_m A^{(m-1)} - A^{(m-1)})^T \\ & + 2W_m (W_m^T W_m A^{(m-1)} - A^{(m-1)}) A^{(m-1)T} \\ & + \xi_2 g'(W_m A^{(m-1)}) A^{(m-1)T} \end{aligned} \quad (3.13)$$

$$\begin{aligned} R^{(m)} = & \left\| W_m^T W_m A^{(m-1)}(Z) - A^{(m-1)}(Z) \right\|_2^2 \\ & + \xi_2 g(W_m A^{(m-1)}(Z)) \end{aligned} \quad (3.14)$$

$$H^{(m)} = W_m A^{(m-1)}(Z) + b_m. \quad (3.15)$$

With the gradients, we iteratively update W_m and b_m until convergence:

$$W_m = W_m - \mu \frac{\partial F}{\partial W_m} \quad (3.16)$$

$$b_m = b_m - \mu \frac{\partial F}{\partial b_m} \quad (3.17)$$

We use $\tanh$ as the activation function t :

$$t(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.18)$$

$$t'(x) = \tanh'(x) = 1 - \tanh^2(x). \quad (3.19)$$

Same as that in JFLE, $g(x)$ is defined as

$$g(x) = \sqrt{\xi^2 + x^2} \quad (3.20)$$

where ξ is a parameter.

What we present above is just one single-layer nonlinear structure (with multiple sub-layers) of the JDFLE. We further stack the single-layer nonlinear structure to have a hierarchical structure to capture multi-level information in RGB-D data, similar to that for JFLE described in section 3.1.3.

3.3 Experiments

3.3.1 Dataset

We use the benchmark dataset - the NYU depth dataset [84, 86], including version 1 (V1) and version 2 (V2), for evaluation. The V1 dataset contains 2347 RGB-D images captured in 64 different indoor scenes labeled with 12 categories plus an unknown class. The V2 dataset consists of 1449 images captured in 464 different scenes.

3.3.2 Implementation details

For each stacked layer, we randomly sample 20000 7×7 patches as training data. We run 50 iterations to learn the feature mapping and dictionary U in our unsupervised learning frameworks for both JFLE and JDFLE.

Figure 3.5: The confusion matrices of: (a) our results with one-layer JFLE structure; (b) our results with two-layer JFLE structure.

In the JFLE framework, the size of W is 200×98 . The parameters λ_1 , λ_2 and λ_3 in Eq. (3.2) are set to 0.1, 0.5 and 0.15. Each iteration takes about 17 minutes on average on a PC with Intel i5 3.10GHz CPU and 8G memory. ξ is set as 10^{-5} for both JFLE and JDFLE.

For JDFLE framework, we pass raw data through 2 nonlinear mapping sub-layers ($M = 2$) for one single layer. The sizes of W_1 and W_2 are set to 98×150 and 150×200 respectively. Two such single-layer structures are stacked as shown in Fig. 3.3, similar to the JFLE framework. The parameters λ , ξ_1 , ξ_2 , ξ_3 in Eq. (3.6) are set to 2, 0.03, 0.1, 0.1. Each iteration takes about 21 minutes on average.

For a superpixel, we calculate the mean values of all its 10×10 patches' features. With the labeled superpixels in the training list, we train a one-vs-all linear SVM for each category. For NYU depth dataset V1 [84], we use 60% data for training and 40% data for testing which is the same as that of [76]. For NYU depth dataset V2 [86], we use the

Table 3.1: Class-average accuracy comparison of different methods on the NYU depth dataset V1.

Single	JFLE	61.71%
	gradient KDES [76]	51.84%
	color KDES [76]	53.27%
	spin/surface normal KDES [76]	40.28%
	depth gradient KDES [76]	53.56%
Combined	Silberman and Fergus [84]	53%
	Pei <i>et al.</i> [73]	50.50%
	Ren <i>et al.</i> [76]	71.40%
	Combining ours with Ren's	72.94%

training/testing splits provided by the dataset: 795 images for training and 654 images for testing.

We produce a confusion matrix whose diagonal elements represent the pixel-level labeling accuracy for each category in Fig. 3.5.

The average value of the diagonal of the confusion matrix is used as the performance metric. Note that different oversegmentation levels lead to different scene labeling results. We report the best performance of different oversegmentation levels. We would also like to point out that in this research we focus on feature learning and thus we did not further apply contextual models such as MRFs to smooth the class labels. For fair comparison, we only report the results of other methods without further smoothing, except the method of Khan *et al.* [45], as the higher-order graphical model is one of their major contributions.

Table 3.2: Class-average accuracy results of our method with different settings on the NYU depth dataset V1.

1	JFLE (first layer)	54.76%
2	JFLE (second layer)	52.90%
3	Sparse coding after feature learning	45.67%
4	k-means encoding after feature learning	22.32%
5	Separate learning from two modalities	50.74%

Table 3.3: Individual class label accuracy on the NYU depth dataset V1 with only one-layer features. The bold numbers are to indicate the cases that extra depth features hurt the performance. In contrast, the performance is boosted when jointly learning from two modalities for all the categories.

	Learning only from color modality	Separate learning from two modalities	Joint learning from two modalities
bed	58.08%	57.11% ↓	62.55%
blind	56.63%	55.19% ↓	60.40%
book	54.88%	47.97% ↓	60.99%
cabinet	34.66%	38.40%	44.77%
ceiling	61.77%	79.52%	75.36%
floor	67.70%	83.20%	81.37%
picture	35.56%	47.35%	51.71%
sofa	43.99%	57.37%	54.48%
table	19.27%	23.68%	30.40%
tv	47.56%	59.83%	73.15%
wall	70.75%	69.73% ↓	71.62%
window	33.69%	36.50%	38.73%
other	3.70%	3.87%	6.28%

Figure 3.6: Visual comparisons of our JFLE and JDFLE results. From left to right: original color images, the results of combining our JFLE features and Ren's features, the results of combining our JDFLE features and Ren's features, ground truth.

Table 3.4: Labeling accuracy on the NYU depth dataset V1 for first-layer, second-layer, both-layer feature learning and feature encoding structure. Second column: the results of our JFLE framework. Third column: the results of our JDFLE framework.

	JFLE	JDFLE
First layer	54.76%	59.46%
Second layer	52.90%	57.18%
Two layers	61.71%	63.28%
Combined with Ren's	72.94%	73.68%

3.3.3 Results

3.3.3.1 Comparisons on dataset V1

Table 3.1 shows the average labeling results of different methods on the NYU depth dataset V1. We compare the result of our two-layer JFLE method with: 1) the result of Silberman and Fergus [84]; 2) the result of Pei *et al.* [73]; 3) the result of single kernel descriptor(KDES) [76]; 4) the result of Ren *et al.* [76] (combining four KDESs and geometry features); 5) the result of combining the features of our JFLE method and Ren's.

It can be seen from Table 3.1 that our method significantly outperforms the method of Silberman and Fergus, as they mainly use SIFT features on color and depth images. Our method also outperforms the method of Pei *et al.* [73], as they use selected patches which are usually redundant and noisy in encoding. However, our result does not outperform that of Ren *et al.* [76]. We argue that Ren *et al.* [76] evaluated six kernel based features, integrated four of them: gradient, color, depth gradient, spin/surface normal. For six kernel descriptors, each of them has several hyperparameters to be tuned. The authors also conducted sophisticated evaluation process to select which of them to be used for classification and finally chose four of them. The weights between these descriptors also need to be tuned to obtain the best combination. In contrast, we just learn a single type of

features directly from raw pixel values. If we compare our result with that of each single descriptor of [76], our method achieves superior performance. Compared to [76], our method does not need any detailed hand-crafting of features. Moreover, by combining our and Ren’s features together, the classification accuracy can be further improved, suggesting that our features capture significant complementary visual patterns which cannot be captured by those of [76].

Evaluating different feature learning and encoding settings: Here, we give detailed evaluations on our linear method with only one-layer feature learning and feature encoding structure under different settings. In particular, we compare the following five setups: 1) our JFLE method with the features learned from the first layer; 2) our JFLE method with the features learned from the second layer alone; 3) separate learning: conducting feature learning with (3.1) to get filter matrix W and then performing sparse coding to encode filtered data WZ ; 4) conducting feature learning with (3.1) and then use k-means clustering result as hard quantization to encode filtered data WZ ; 5) learning features from two modalities separately with our JFLE cost function. Table 3.2 shows the results of the five different setups.

Comparing the results of setups 1, 3 and 4 in Table 3.2, we can see that joint feature learning and encoding performs much better than the setups using separate processing. Particularly, for setup 3, we run 50 iterations to update W and then conduct sparse coding for 50 iterations to encode WZ . Compared with the way of iteratively updating W and U for 50 iterations in setup 1, setup 3 cannot guide W to help find descriptive U . For setup 4, important feature information is lost when quantized by k-means.

Comparing the results of setups 1 and 5 in Table 3.2, we can see that joint learning from two modalities outperforms separate learning. This is because separate learning ignores the complimentary information between the two modalities, for which the extra features

learned from depth alone might hurt the performance, as shown in Table 3.3. On the contrary, our algorithm implicitly infers the correlation between the two modalities and could find better combination of them, which leads to better performance for all the classes (see Table 3.3).

Comparing the results of setups 1 and 2 in Table 3.2, we can see that the high-level features captured by the second layer alone are not sufficient. Only when combining with low-level features together, we can achieve a performance improvement of 7% (see Table 3.1), compared with using the first-layer features alone. The detailed comparison of confusion matrices between one-layer JFLE and two-layer JFLE is shown in Fig. 3.5. For individual classes: “bed”, “blind”, and “window”, low level visual information e.g. intensity, edge is not enough to describe objects in these classes. We can observe that performances of these classes are significantly improved by combining high-level features.

Efficacy of JDFLE: To illustrate the efficacy of the JDFLE framework, the comparison with results of the JFLE framework on NYU dataset V1 is shown in Table 3.4. We can see that the JDFLE framework improves the performance of first-layer, second-layer, and two-layer structures compared to JFLE. This is because we learn more general model for data through several nonlinear projections, and the simultaneously learned dictionary becomes more efficient to encode the data. In Fig. 3.6, several visual examples are shown to illustrate the improvement made by JDFLE framework.

3.3.3.2 Comparisons on dataset V2

We also compare our results on NYU depth dataset V2 with the following existing works that have reported results on the dataset V2: 1) Couprie *et al.* [15], which automatically learns features from raw data input, similar to our method; 2) Cadena and Kosecka [12];

3) Khan *et al.* [45]. [12] and [45] both include a lot of hand-crafted appearance and geometry features. Table 3.9, 3.10, 3.11 show the individual class labeling accuracy results on the NYU depth dataset V2 with the same 4-class, 13-class, 40-class settings. For 40-class setting, we also show the Jaccard index of each class. The overall performance summaries are shown in Table 3.5 and 3.6 including: overall pixel-level accuracy (Pix. Acc.) and average pixel-level labeling accuracy for all classes (Per-Class Acc.) for 4-class, 13-class, 40-class settings, the mean pixel-frequency weighted Jaccard index (Freq. Jaccard), and the flat mean Jaccard index (Av. Jaccard) for the 40-class setting. It can be seen that our JFLE and JDFLE methods in general achieve the competitive accuracy compared to [15], [12] and [45]. We would like to point out that a very recent work [21] by Eigen and Fergus reported a better performance on NYU depth dataset V2, e.g. its pixel-level accuracy for 40-class setting has reached 62.9%. However, their method is based on CNN with supervised information required, while we use unsupervised feature learning, i.e. deriving features without using any label information. A direct comparison with their method is unfair as the problem settings are different. Compared with JFLE, JDFLE also achieves improved performance on NYU depth dataset V2. Table 3.7 shows the comparison between 4-class average class accuracy performance of JFLE and JDFLE with first-layer, second-layer, both-layer structures.

Fig. 3.8 and Fig. 3.9 show some examples of pixel labeling results on NYU depth dataset V1 and V2 respectively. The visualization results demonstrate that the learned local features can well represent objects in the scene. Note that since our work focuses on feature learning, we did not use CRFs or MRFs to smooth class labels.

[‡]Khan *et al.* [45] used a different label set. See the details in Table 3.10.

Table 3.5: Labeling accuracy of 4-class and 13-Class settings on the NYU depth dataset V2.

4-Class	Pix. Acc.	Per-Class Acc.
Coupric <i>et al.</i> [15]	64.5	63.5
Cadena and Kosecka [12]	64.9	64.1
Khan <i>et al.</i> [45]	69.2	65.6
JFLE	67.8	65.3
JDFLE	69.7	66.9
13-Class	Pix. Acc.	Per-Class Acc.
Coupric <i>et al.</i> [15]	52.4	36.2
Khan <i>et al.</i> [45]	‡	
JFLE	47.4	42.2
JDFLE	52.8	44.1

Table 3.6: Labeling accuracy of 40-class on the NYU depth dataset V2.

40-Class	Pix. Acc.	Per-Class Acc.	Freq. Jaccard	Av. Jaccard
JFLE	48.9	27	35.8	17.4
JDFLE	51.6	29.2	38.1	19

Table 3.7: Labeling accuracy on the NYU depth dataset V2 for first-layer, second-layer, both-layer feature learning and feature encoding structure using our proposed JFLE and JDFLE frameworks.

	JFLE	JDFLE
First layer	60.2%	62.4%
Second layer	58.7%	61.3%
Two layers	65.3%	66.9%

Figure 3.7: Scene labeling accuracy under different λ values with single-layer JDFLE structure on NYU depth dataset V1.

Table 3.8: The results of different settings of JDFLE.

	Accuracy
Two nonlinear sub-layers	59.46%
One nonlinear sub-layers	58.21%

3.3.3.3 Parameter analysis

Different settings of JDFLE: Different nonlinear structures could be constructed to learn the nonlinear mapping of the input RGB-D data. In Table 3.4, we show the result with two nonlinear sub-layers (i.e. $M = 2$) to map the data. We also test more shallow network with $M = 1$. The results are shown in Table 3.8. It can be seen that with only one nonlinear sub-layer, the result is worse than that of two sub-layers. We did not see the performance improvement using more than two nonlinear sub-layers.

Effect of different λ : Fig. 3.7 shows the accuracy results under different λ in Eq. (3.6)

Table 3.9: Individual class labeling accuracy of 4-Class setting on the NYU depth dataset V2.

	Ground	Furniture	Props	Structure
Couprie <i>et al.</i> [15]	87.3	45.3	35.5	86.1
Cadena and Kosecka [12]	87.3	60.6	33.7	74.8
Khan <i>et al.</i> [45]	87.1	54.7	32.6	88.2
JFLE	90.1	46.3	43.3	81.4
JDFLE	87.3	50.7	47.4	82.2

Table 3.10: Individual class labeling accuracy of 13-Class setting on the NYU depth dataset V2.

	bed	objects	chair	furnit.	ceiling	floor	deco.	sofa	table	wall	window	books	tv
Couprie <i>et al.</i> [15]	38.1	8.7	34.1	42.4	62.6	87.3	40.4	24.6	10.2	86.1	15.9	13.7	6
Khan <i>et al.</i> [45]	32.3	-	-	-	64.7	75.8	-	58.6	47.9	77.5	54	38.3	45.7
JFLE	47.6	12.4	23.5	16.7	68.1	84.1	26.4	39.1	35.4	65.9	52.2	45	32.4
JDFLE	48.5	19.5	26.2	24.8	63.2	89.5	27.3	37	36.3	76.4	49.3	43.2	33.1

with single-layer JDFLE structure on NYU depth dataset V1. Recall that λ denotes the tradeoff between feature learning and feature encoding. If λ is too small, feature encoding becomes dominating while weak feature learning will not be able to learn discriminative features. On the other hand, if λ is too big, feature learning becomes dominate while weak feature encoding will not be able to produce effective sparse representation. Fig. 3.7 clearly shows such a tradeoff, which suggests an optimal λ value around 2.

3.4 Summary

In this chapter, we describe our method for RGB-D scene labeling task. In particular, we propose *joint feature learning and encoding* framework (JFLE), where we optimize feature learning and feature encoding jointly. We stack learning framework to learn hierarchical features. Extensive empirical studies have been conducted on NYU depth dataset V1 and V2, which validate the efficacy of JFLE. To make the learning process

JACCARD INDEX	wall	floor	cabinet	bed	chair	sofa	table	door	window	book shelf	picture	counter	blinds	desk	shelves	curtain	dresser	pillow	mirror	floor mat
JFILE	60.1	64.1	27.5	46.1	29	29.4	20.2	6.5	23.6	13.3	24.8	31.2	31	4.3	1.1	16	15.1	11.8	9	14.7
JDFLE	61.4	66.4	38.2	43.9	34.4	33.8	22.6	8.3	27.6	17.6	27.7	30.2	33.6	5.1	2.7	18.9	16.8	12.5	10.7	13.8
	clothes	ceiling	books	fridge	television	paper	towel	shower curtain	box	whitebd	person	night stand	toilet	sink	lamp	bathub	bag	other struct	other furnit	other prop
JFILE	2.7	46.1	3.6	2.9	3.2	2.6	6.2	6.1	0.8	28.2	5	6.9	32	20.9	5.4	16.2	0.2	4.8	1.1	21.7
JDFLE	3.3	47.9	2.5	4.3	5.8	4	6.4	1	1.3	19.2	6	8.9	33.3	21.7	6.7	26.5	0.7	5.6	2.7	24.6
PIXEL ACCURACY	wall	floor	cabinet	bed	chair	sofa	table	door	window	book shelf	picture	counter	blinds	desk	shelves	curtain	dresser	pillow	mirror	floor mat
JFILE	76.7	87.1	35.3	61	41.7	44.9	29.9	10.3	43.9	24.6	41.7	51.4	47.4	8.4	2.7	27	22.1	24.6	19.5	25.8
JDFLE	77.4	89.1	48.7	57.2	48.7	50.9	32.8	12.8	49.9	31.8	45.6	48.8	50.2	9.7	6.6	31	24.1	25.5	22.6	23.5
	clothes	ceiling	books	fridge	television	paper	towel	shower curtain	box	whitebd	person	night stand	toilet	sink	lamp	bathub	bag	other struct	other furnit	other prop
JFILE	8.2	69.6	9.2	6.3	5.1	7.6	15.1	7.5	3.6	32.4	14.2	11.9	56.4	32.6	20.4	18.2	0.4	8.1	1.8	25.4
JDFLE	9.6	71	6.3	8.9	8.9	11.4	14.9	1.3	5.7	21.9	16.2	15	56.9	33.4	24.5	29.6	2.1	9.2	4.3	28.6

Table 3.11: Individual class labeling accuracy of 40-Class setting on the NYU depth dataset V2.

Figure 3.8: 15 example results on NYU depth dataset V1. Rows 1st, 4th and 7th: color images. Rows 2nd, 5th and 8th: the results of combining our JDFLE features and Ren's features. Rows 3rd, 6th and 9th: ground truth. Note that since we focus on feature learning, we did not use CRFs to smooth the labels. So the results might look a bit noisy.

Figure 3.9: 12 example results on NYU depth dataset V2. Rows 1st and 4th: color images. Rows 2nd and 5th: the results of our JDFLE features. Rows 3rd and 6th: ground truth.

more general, we further extend JFLE to *joint deep feature learning and encoding* (JD-FLE), where nonlinear mapping is used instead of linear projection.

Chapter 4

MMSS: Multi-modal Sharable and Specific Feature Learning for RGB-D Object Recognition

To exploit the implicit dependence between different modalities, we propose a multi-modal learning framework for RGB-D object recognition that treats color and depth as two modalities. At the heart of our method, we jointly explore two kinds of feature properties: shared common patterns of different modalities, and modal-specific patterns owned by individual modalities. We learn a compact and discriminative representation by transforming the data of each modality to a new feature domain with two parts: the common feature part shared by all modalities and the modal-specific part. By concatenating the shared and modal-specific features from all modalities, we obtain the final object representation which has the intended desirable properties. Supervised information is further integrated into the framework to enhance discriminative capability. CNN layers are constructed to form the input to our multi-modal feature learning framework, and the information of the multi-modal learning framework is back-propagated to the early CNN layers. The multi-modal feature learning and the back-propagation are iteratively performed until convergence. Fig. 4.1 shows the structure of the proposed multi-modal feature learning framework.

Figure 4.1: Illustration of our proposed multi-modal feature learning framework. The inputs X and Y are the activations of the second fully connected layers of the CNNs pre-trained on color and depth separately. The inputs are transformed by W_1 and W_2 respectively, and the transformed features T_1 and T_2 are enforced to share a common part T_c . The labeling information L is integrated in the learning process to enhance the discrimination.

Figure 4.2: Different CNN structures for RGB-D data. Green, blue, yellow and red boxes indicate convolutional, pooling, fully-connected and softmax layers, respectively.

4.1 Proposed Approach

4.1.1 Conventional CNN-based Learning Structure

Of the various ways of using CNNs with RGB-D data, a straightforward approach is to combine RGB and depth data from the outset as a four-channel input to the convolutional neural network, as shown in Fig. 4.2(a) (akin to what is used in Couprie *et al.* [15] for scene labeling), where green, blue, yellow and red boxes indicate convolutional, pooling, fully-connected and softmax layers, respectively. Alternatively, discriminative features can be extracted independently from color and depth images by concatenating the activations of the second fully-connected layers of the two modalities, and feeding them into the last fully-connected layer with dense connections. From the final softmax layer, supervised information is back-propagated to the independent networks for both modalities. Such a structure is shown in Fig. 4.2(b).

4.1.2 Proposed Multi-modal Learning Structure

Rather than adopting the above two conventional learning structures that involve some simple fusions of color and depth data, we propose to explore the relationship between the two modalities. We develop an architecture for multi-modal feature learning carried out in conjunction with convolutional neural networks. Specifically, we first pre-train CNNs on color and depth images separately, as shown in Fig. 4.2(c) and (d). Then, the activations of the second fully-connected layers of the two modalities are fed into the proposed multi-modal feature learning framework shown in Fig. 4.1.

Our main idea is that the desired features should reflect the agreement or shared properties between different modalities, while at the same time they should contain the modal-specific properties that are only captured by one of the modalities. To realize such a goal, we explicitly enforce the learned features of different modalities to share a common part. In addition, the weights between different modalities are simultaneously learned in our framework without prior knowledge of which modality is the most important one.

In Fig. 4.1, $X = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{M_1 \times N}$ denotes the activations (with M_1 dimensions) of the second fully-connected layer of the CNN for color images in one data batch with N images. Similarly, $Y \in \mathbb{R}^{M_2 \times N}$ denotes the activations of the CNN for depth images in one data batch. Our objective is to learn a new feature representation T containing two sets of properties: 1) common properties shared by two modalities; and 2) modal-specific properties captured separately by individual modalities.

Let $T_1 \in \mathbb{R}^{M_1' \times N}$ and $T_2 \in \mathbb{R}^{M_2' \times N}$ denote the learned features for the color and depth modalities respectively. Here we enforce T_1 and T_2 to 1) share a common part $T_c \in \mathbb{R}^{K_c \times N}$, and 2) contain modal-specific parts $T_{1s} \in \mathbb{R}^{K_{1s} \times N}$ and $T_{2s} \in \mathbb{R}^{K_{2s} \times N}$ respectively, where $M_i' = K_c + K_{is}$, $i = 1, 2$, as illustrated in Fig. 4.3. The learned features for the two

Figure 4.3: Illustration of the transformation matrix.

Figure 4.4: Illustration of the regression coefficient matrix.

modalities are therefore: $T_1 = [T_{1s}; T_c]$ and $T_2 = [T_{2s}; T_c]$. We further denote $W_i \in \mathbb{R}^{M_i' \times M_i}$ as the transformation matrix for modality i . Our task is then to learn the appropriate transformation matrices W_i to obtain the features $T_1 = W_1 X$ and $T_2 = W_2 Y$. The final learned features are $T = [T_c; T_{1s}; T_{2s}]$.

Finally, we require an additional matrix $W \in \mathbb{R}^{(K_c + K_{1s} + K_{2s}) \times N_c}$ to map the feature T representation to actual labels for N_c number of classes, as illustrated in Fig. 4.4. Here we incorporate supervised learning by enforcing $W^T T$ to be close to the ground truth label L .

4.1.3 Formulation

To learn features containing both shared and modal-specific properties, we formulate our cost function as

$$\begin{aligned}
 & \min_{\{W_1, W_2, \alpha_1, \alpha_2, T_1, T_2, W\}} F = F_1 + F_2 + F_3 \\
 & = \alpha_1 (\|W_1 X - T_1\|_F^2 + \|W_1^T T_1 - X\|_F^2 + \lambda_1 g(T_1)) \\
 & + \alpha_2 (\|W_2 Y - T_2\|_F^2 + \|W_2^T T_2 - Y\|_F^2 + \lambda_1 g(T_2)) \\
 & + \beta (\|W^T T - L\|_F^2 + \lambda_2 \|W\|_{2,1}) \\
 & \text{subject to } \alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0
 \end{aligned} \tag{4.1}$$

where $\|\cdot\|_F$ and $\|\cdot\|_{2,1}$ denote the Frobenius norm and $l_{2,1}$ norm. F_1 is the cost for regulating the color modality, in which: the first term enforces T_1 to be similar to the W_1 -transformed X , the second term encourages the ability of T_1 to reconstruct X when back-transformed via W_1^T , while the third term g is the smooth L_1 penalty function [52]. Likewise, F_2 corresponds to the cost for regulating depth modality. α_1, α_2 are weights for color and depth modalities. λ_1 and λ_2 are trade-off parameters for regularization. β is the trade-off parameter for F_3 . Although the definitions of F_1 and F_2 seem to indicate that color and depth modalities are optimized independently, in fact T_1 and

T_2 are not independent since they are explicitly required to share a common part T_c . By concatenating T_c and the modal-specific parts T_{1s} , T_{2s} , the final representation for each image is $T = [T_c; T_{1s}; T_{2s}]$. The third part, F_3 in (4.1), incorporates supervised information to enhance the discriminative power of the learned features, in which W is the regression coefficient matrix and the $l_{2,1}$ norm ensures W to be row-wise sparse, thus acting as a feature selector in T . Fig. 4.3 illustrates the matrix transformation in F_1 , while Fig. 4.4 shows the regression coefficient matrix in F_3 .

After we derive the learned matrices W , W_1 and W_2 in the training stage, the features of any test image can be directly computed as: $T_{1s} = W_{1s}X$, $T_{2s} = W_{2s}Y$, $T_c = (W_{1c}X + W_{2c}Y)/2$. With the multi-modal feature representation $T = [T_c; T_{1s}; T_{2s}]$, the final recognition result will be directly computed as $W^T T$.

4.1.4 Alternating Optimization

In this research, we employ the typical alternating optimization strategy to obtain a local optimal solution for (4.1). The pipeline of the algorithm is briefly described in Alg. 2. First, W , W_1 , W_2 and T are initialized randomly, and α_i is initialized as 0.5. All these variables including W , W_i , T and α_i will be learned and updated in Alg. 2. Other parameters such as λ_1 , λ_2 and β in (4.1) are set via cross validation.

In Step 2.1, we fix W_i , W , T , and update α_i . α_1 and α_2 allow the different modalities to have different weights since they are unlikely to play the same role. When W_i , W , T are fixed, we can construct the following Lagrange function based on (4.1):

$$L(\alpha, \eta) = \alpha_1 C_1 + \alpha_2 C_2 + \beta C - \eta(\alpha_1 + \alpha_2 - 1). \quad (4.2)$$

where C_1 , C_2 and C are the corresponding constant values in (4.1) due to fixing W_i , W , T . η is the Lagrange multiplier. β is trade-off parameter for F_3 of our cost function.

Algorithm 2: Optimizing the proposed multi-modal feature learning framework

Input: Training set with two modalities: X, Y , the corresponding ground truth label L .

Output: Feature projection matrix: W_1, W_2 . Regression coefficient matrix W .

Step 1 (Initialization):

Initialize $W, W_1, W_2, T, \alpha_1, \alpha_2$.

Step 2 (Optimization):

loop

2.1. Fix W, W_1, W_2, T

Update α_1, α_2 according to (4.5).

2.2. Fix $W_1, W_2, T, \alpha_1, \alpha_2$,

Update W according to (4.7).

2.3. Fix $W, W_1, W_2, \alpha_1, \alpha_2$,

Update T_c and T_{is} according to (4.10).

2.4. Fix W, T, α_1, α_2 ,

Update W_1, W_2 according to (4.12).

end loop until convergence

Unfortunately, the solution to (4.2) will be trivial. For example, if C_1 is less than C_2 , then the solution of minimizing (4.2) will be: $\alpha_1 = 1$ and $\alpha_2 = 0$, which means only one modality will be used in the feature learning. Experimentally we found that this leads to suboptimal results. In order to utilize the information from different modalities, we modify our cost function to

$$\begin{aligned}
 \min_{\{W_1, W_2, \alpha_1, \alpha_2, T_1, T_2, W\}} F &= F_1 + F_2 + F_3 \\
 &= \alpha_1^p (\|W_1 X - T_1\|_F^2 + \|W_1^T T_1 - X\|_F^2 + \lambda_1 g(T_1)) \\
 &+ \alpha_2^p (\|W_2 Y - T_2\|_F^2 + \|W_2^T T_2 - Y\|_F^2 + \lambda_1 g(T_2)) \\
 &+ \beta (\|W^T T - L\|_F^2 + \lambda_2 \|W\|_{2,1}) \\
 \text{subject to } &\alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0
 \end{aligned} \tag{4.3}$$

where $p > 1$ is an additional parameter. By adding p , the objective becomes nonlinear for α_i and the two modalities will be constrained to obtain shared common pattern and modal-specific patterns in T , while at the same time keeping the most of the original information in T . In this way, the Lagrange function becomes

$$L(\alpha, \eta) = \alpha_1^p C_1 + \alpha_2^p C_2 + \beta C - \eta(\alpha_1 + \alpha_2 - 1). \tag{4.4}$$

By setting $\frac{\partial L(\alpha, \eta)}{\partial \alpha}$ and $\frac{\partial L(\alpha, \eta)}{\partial \eta}$ to 0, α_i can be updated as:

$$\alpha_i = \frac{(1/C_i)^{1/(p-1)}}{\sum_{i=1}^2 (1/C_i)^{1/(p-1)}}. \quad (4.5)$$

In Steps 2.2-2.4, we update the other variables using the gradient descent algorithm, where the same learning rate γ is used. In particular, the regression coefficient matrix W is updated in Step 2.2. According to [71], the derivative of the cost function with respect to W can be expressed as

$$\frac{\partial F}{\partial W} = 2\beta(T(W^T T - L)^T + \lambda_2 E W) \quad (4.6)$$

where E is a diagonal matrix with $e_{kk} = 1/2\|w_k\|_2$, and w_k is the k th row of W . Then, W is updated according to the gradient descent rule:

$$W \leftarrow W - \gamma \frac{\partial F}{\partial W}. \quad (4.7)$$

In Step 2.3, the feature representation T is updated. Considering that T contains a common part T_c and modal-specific parts T_{1s} and T_{2s} , we update these three parts separately. In this way, the learned features contain both shared common properties and modal-specific properties. The derivatives of F with respect to T_c and T_{1s} , and the mechanism for updating T_c and T_{1s} (and likewise for T_{2s}) are shown below:

$$\begin{aligned} \frac{\partial F}{\partial T_c} = & 2\alpha_1^p [(T_c - W_{1c}X) + W_{1c}(W_{1c}^T T_c - X) + \lambda_1 g'(T_c)] \\ & + 2\alpha_2^p [(T_c - W_{2c}Y) + W_{2c}(W_{2c}^T T_c - Y) + \lambda_1 g'(T_c)] \\ & + 2\beta W^{(c)}(W^{(c)T} T_c - L) \end{aligned} \quad (4.8)$$

$$\begin{aligned} \frac{\partial F}{\partial T_{1s}} = & 2\alpha_1^p [(T_{1s} - W_{1s}X) + W_{1s}(W_{1s}^T T_{1s} - X) + \lambda_1 g'(T_{1s})] \\ & + 2\beta W^{(1s)}(W^{(1s)T} T_{1s} - L). \end{aligned} \quad (4.9)$$

The common part and the modal-specific parts of T are updated according to the gradient descent rule:

$$T_c \leftarrow T_c - \gamma \frac{\partial F}{\partial T_c} \quad T_{1s} \leftarrow T_{1s} - \gamma \frac{\partial F}{\partial T_{1s}}. \quad (4.10)$$

In Step 2.4, when T , W and α_i are fixed, W_i is updated in a similar way, i.e.,

$$\frac{\partial F}{\partial W_1} = 2\alpha_1^p \left[(W_1 X - T_1) X^T + T_1 (W_1^T T_1 - X)^T \right] \quad (4.11)$$

$$W_1 \leftarrow W_1 - \gamma \frac{\partial F}{\partial W_1}. \quad (4.12)$$

In our framework, X and Y are the activations of the second fully-connected CNN layers. The results of the multi-modal learning will then be back-propagated to the lower layers of CNN by

$$\frac{\partial F}{\partial X} = 2\alpha_1^p \left[W_1^T (W_1 X - T_1) - (W_1^T T_1 - X) \right]. \quad (4.13)$$

The multi-modal feature learning and the back-propagation are iteratively performed until convergence. Here, we have shown the formulation for a two-modal problem. It is straightforward to extend to a multi-modal formulation by representing the learned features with the concatenation of the common part and the modal-specific parts from additional modalities.

4.2 Experiments

To evaluate the effectiveness of our proposed multi-modal feature learning framework, we perform object recognition experiments on the RGB-D Object Dataset [51] and the 2D3D Dataset [11]. The details of the experiments and the results are described in the following subsections.

4.2.1 Datasets and Experiment Setup

RGB-D Object Dataset: This dataset has 51 object classes and contains RGB-D images of 300 distinct objects taken from multiple views. They are commonplace objects such as cups, keyboards, fruits and vegetables. Each object is video-recorded with cameras mounted at three different elevation angles of approximately 30° , 45° and 60° . There are in total 207,920 RGB-D image frames, with roughly 600 images per object.

We conduct experiments for both category recognition and instance recognition. We adopt the same setup as [10], where images are sampled from every 5th frame of the videos. For the category recognition, we run the 10 random splits provided. For each split, one object from each class is sampled, resulting in 51 test objects. There are some 34,000 images for training and 6900 images for testing. For the instance recognition, we use images captured from elevation angles of 30° and 60° for training, and test on the images of the 45° angle (leave-sequence-out).

2D3D Dataset: This dataset consists of 154 objects in 14 different classes. Each object is recorded by a *PMDTM* CamCube 2.0 time-of-flight camera with views at every 10° around the vertical axis, resulting in a total of 5544 RGB-D images. For category recognition, we adopt the setting of [11]. After excluding some classes with few examples, 6 objects of each class are used for training and the remaining objects are used for testing. For each training (testing) object, only 18 views out of 36 views are used. Eventually 82 objects in 1476 RGB-D images are regarded as training data, while 74 objects in 1332 RGB-D images are used for testing.

Architecture of CNNs: As we consider two modalities, for each modality we construct a smaller network than the one in [49] in order to ensure that the data of the two modalities can be placed in the GPU memory simultaneously. The input images are resized

to 150×150 . For the color modality, there are 96 kernels of size $7 \times 7 \times 3$ with stride 2, 96 kernels of size $5 \times 5 \times 96$ with stride 2, 112 kernels of size $3 \times 3 \times 96$ with stride 1, 128 kernels of size $3 \times 3 \times 112$ with stride 1, and 128 kernels of size $3 \times 3 \times 128$ with stride 1, for the filters of the 1st, 2nd, 3rd, 4th and 5th convolutional layers, respectively. The two fully-connected layers have the sizes of 1024 and 512 respectively. A dropout of 0.5 probability is used for the first fully-connected layer. For each 150×150 image, overlapping 142×142 images are cropped for data augmentation. There are max-pooling layers following the first, the second and the fifth convolutional layers. ReLu non-linearity [49] is applied to the output of every convolutional layer and every fully-connected layer. Note that when initializing CNNs by independently training with color and depth images as shown in Fig. 4.2(c) and (d), the final fully-connected layer has a size equal to the number of categories, which is then fed into the final softmax layer. We use the same architecture for both color and depth modalities, apart from the size of filters in the first convolutional layer (3 channels for color and 1 channel for depth).

Parameters setting: For our multi-modal learning framework, the dimension M_k' of the transformed features is set to be the same as $M_k = 512$, although it could be different. Half of the M_k' -dimensional features are enforced to be the same between the two modalities, i.e. $K_c = 256$ and $K_{is} = 256$. The parameters $p, \beta, \lambda_1, \lambda_2, \gamma$ are set as 2, 1000, 1, 20, 0.001 respectively via cross validation for all the experiments for both the RGB-D object and 2D3D datasets.

4.2.2 Results on RGB-D Object Dataset

Comparison with different baselines of using CNNs: We compare with five different CNN-based baselines: 1) CNN trained using RGB images only (Fig. 4.2(c)), named ‘RGB CNN’; 2) CNN trained using depth images only (Fig. 4.2(d)), named ‘Depth

CNN'; 3) RGB-D used as the four-channel input to a CNN (Fig. 4.2(a)), named 'RGB-D CNN with 4-channel input'; 4) CNN with separate training for color and depth at the lower layers, followed by concatenating the activations of the second fully-connected layer (*fc7*) and feeding them into the last fully-connected layer (Fig. 4.2(b)), named 'RGB-D CNN connected at *fc7*'; 5) Similar setting with 4), but two modalities are concatenated at the fifth convolutional layer (*conv5*), named 'RGB-D CNN connected at *conv5*'.

The top part of Table 4.1 shows the recognition results of the five baselines on RGB-D Object Dataset. It can be seen that although simply adding depth as the fourth channel of the CNN input ('RGB-D CNN with 4-channel input') greatly improves the performance of those only using one modality ('RGB CNN' and 'Depth CNN'), extracting features separately from color and depth and connecting them at the later stage ('RGB-D CNN connected at *conv5*') performs better with significant gain. This is because separately learning features at the early stage for different modalities result in more independent features, which could prevent the CNN from primarily learning features for the predominant modality.

Following [10], we also use surface normals to replace the depth map as the input, which results in another three baselines: 6) CNN trained using surface normals only, named 'Surface Normals (SN) CNN'; 7) RGB and surface normals used as the six-channel input to a CNN, named 'RGB-SN CNN with 6-channel input'; 8) CNN with separate training for color and surface normals at the lower layers, followed by concatenating the activations of the second fully-connected layer and feeding them into the last fully-connected layer, named 'RGB-SN CNN connected at *fc7*'; 9) Similar setting with 8), but two modalities are concatenated at *conv5*, named 'RGB-SN CNN connected at *conv5*'. The comparison in Table 4.1 indicates that surface normals can better represent

Table 4.1: Comparison of different baselines of using CNNs on RGB-D Object Dataset.

Method	Accuracy (%)
RGB CNN	74.6 ± 2.9
Depth CNN	75.5 ± 2.7
RGB-D CNN with 4-channel input	80.2 ± 1.9
RGB-D CNN connected at <i>fc7</i>	84.7 ± 2.1
RGB-D CNN connected at <i>conv5</i>	85.1 ± 2.0
Surface Normal (SN) CNN	76.3 ± 2.5
RGB-SN CNN with 6-channel input	80.7 ± 2.1
RGB-SN CNN connected at <i>fc7</i>	85.0 ± 2.4
RGB-SN CNN connected at <i>conv5</i>	85.5 ± 2.2
RGB-SN CNN connected at <i>conv5</i> (pretrained)	86.8 ± 2.1

geometry information than the depth map. To further boost the performance, we use images of first 50 classes of ILSVRC2012 [77] to pretrain CNN layers of both color and surface normals, which leads to another baseline: 10) named ‘RGB-SN CNN connected at *conv5* (pretrained)’, achieving the best performance among all the baselines.

Table 4.2 compares the results of our proposed multi-modal learning with the best baseline, ‘RGB-SN CNN connected at *conv5* (pretrained)’. Note that our method also uses surface normals to replace depth images and uses pretrained CNN layers. It can be seen from Table 4.2 that our method outperforms the best baseline by 1.7% in recognition accuracy. This is mainly because our method extracts both shared common patterns and modal-specific patterns of different modalities, which cannot be achieved through simply connecting color and surface normals by a fully-connected layer.

Comparison with state-of-the-art methods: We also compare our method with state-of-the-art methods including: 1) Lai *et al.* [51]: using SIFT and spin images for depth, and SIFT, color histogram and texon histogram for color; 2) Blum *et al.* [7]: using con-

Table 4.2: Comparison of our method with the best baseline. Here the CNNs in both methods are pretrained with a subset of ILSVRC2012 dataset.

Method	Accuracy (%)
RGB-SN CNN connected at <i>conv5</i> (pretrained)	86.8 ± 2.1
Ours	88.5 ± 2.2

Table 4.3: Comparison with state-of-the-art methods on RGB-D Object Dataset.

Method	Category (%)	Instance (%)
Lai <i>et al.</i> [51]	81.9 ± 2.8	73.9
Blum <i>et al.</i> [7]	86.4 ± 2.3	90.4
Socher <i>et al.</i> [88]	86.8 ± 3.3	-
Zhang <i>et al.</i> [111]	-	86.6
Bo <i>et al.</i> [10]	87.5 ± 2.9	92.8
Ours	88.5 ± 2.2	94.0

volutional k-means descriptors; 3) Socher *et al.* [88]: using Recursive Neural Network plus CNN; 4) Zhang *et al.* [111]: using transfer learning based method; 5) Bo *et al.* [10]: using sparse coding based feature learning with additional input channels such as gray-scale image and surface normals. The comparison results are shown in Table 4.3. It can be seen that our method achieves the best performance, outperforming state-of-the-art method in both category recognition and instance recognition.

The confusion matrix of our final results is shown in Fig. 4.5, whose diagonal elements represent the recognition accuracy for each category. Fig. 4.6 shows a few misclassification examples. For instance, in Fig. 4.6 (a) the light bulb is misclassified as a cap due to similar geometrical shapes. Likewise, the tomato is misclassified as a potato due to strong similarities in both color and shape. In Fig. 4.6 (b), the cellphone is misclassified as toothbrush as there is a missing part of the surface normals, which is misleading.

Figure 4.6: Misclassification examples. Misclassifications are due to similar color, texture or geometry shape.

4.2.3 Results on 2D3D Dataset

We use the CNNs pretrained as described in Sec 4.2.2 and fine-tuned on RGB-D Object Dataset for the initialization. Table 4.4 shows the comparison between our method and the best baseline approach, ‘RGB-SN CNN connected at *conv5* (pretrained)’. This table also shows the comparison results of our method and state-of-the-art methods on this dataset, including Bo *et al.* [10] and Browatzki *et al.* [11], which use multiple descriptors such as 3D shape context and depth buffer for depth and multiple descriptors such as SURF and self similarity features for color. Similar remarks as those for the results on RGB-D Object Dataset can be made here, i.e. our multi-modal feature learning method achieves superior performance to state-of-the-art methods. The confusion matrix is shown in Fig. 4.7.

Figure 4.7: Confusion matrix of the category recognition results on 2D3D Dataset.

Table 4.4: Comparison on 2D3D Dataset.

Method	Accuracy (%)
Browatzki <i>et al.</i> [11]	82.8
Bo <i>et al.</i> [10]	91.0
RGB-SN CNN connected at <i>conv5</i> (pretrained)	89.2
Ours	91.3

Figure 4.8: The effect of choosing different R .

4.2.4 Parameter Analysis

In our method, there are some important parameters. One is

$$R = \frac{K_c}{K_c + K_{is}} \quad (4.14)$$

which ranges from 0 to 1 and controls the percentage of the shared features occupying the transformed features (note that $K_{1s} = K_{2s}$ in our setting). Fig. 4.8 shows how the category recognition performance on RGB-D Object Dataset split 1 varies with different R . When R is too small, the recognition accuracy is relatively low since there is only a small portion of common features between the two modalities, which cannot fully exploit the shared properties. On the other hand, when R is too large, the modal-specific features will vanish.

Another important parameter is β , which balances the relation between the feature re-

Figure 4.9: The effect of choosing different β .

construction constraints and the supervised constraints. Fig. 4.9 shows the accuracy performance under different β values on RGB-D Object Dataset. It can be seen that an excessively small or large weight will result in a performance drop, especially when a small weight is set on the supervised cost.

4.3 Summary

In this chapter, we describe our method for RGB-D object recognition task, where we treat RGB and depth as two different modalities. This method is motivated by the intuition that different modalities should contain both modal-specific patterns and common patterns shared by different modalities. We construct CNNs for different modalities and connect them with our carefully-designed multi-modal layers. We have conducted ex-

tensive experiments on RGB-D Object dataset and 2D3D dataset. The results show that our method outperforms the state-of-the-art methods.

Chapter 5

Modality and Component Aware Feature Fusion for RGB-D Scene Classification

For scene classification task, we start with the standard pipeline for local feature extraction and feature encoding. In particular, we use an existing object proposal extractor to generate region proposals from each RGB-D image, representing each proposal by the corresponding local CNN features obtained from different modalities. Similar to [31], we extract CNN features from RGB and HHA (horizontal disparity, height above ground, and angle between the local surface normal and direction of inferred gravity). In order to more explicitly capture geometric information, we also extract CNN features from an additional modality of *surface normals* (SN). For each modality, we use FV to encode the region-proposal-based CNN features.

As for FV, only a small subset of features are likely discriminative for scene classification task. In addition, the relation between different modalities should be well exploited. To address these two issues, we make two important postulates: Firstly, we consider component sparsity: we should *not* attempt to utilize all features in the FV GMM components, but rather seek out only a few key components that maximally con-

Figure 5.1: Our framework: we first extract proposals from each RGB-D image. Then for all the proposals and the full image, we derive CNN features from different modalities: RGB, HHA and surface normal (SN). For each image, the proposal based CNN features for each modality are encoded by Fisher Vector (FV), and the resulted multi-modal FV features are regarded as the input to our modality and component aware feature fusion. Finally we combine the regression results of the proposal based FV features and the full-image based CNN features to get the final classification result.

tribute to scene discriminability. Secondly, modal non-sparsity is considered: for these key discriminative components, all modalities will significantly contribute to the discriminability because they provide important complementary information.

To this end, we propose a modality and component aware feature fusion framework for RGB-D scene classification on the extracted multi-modal FV features. In the feature fusion step, we incorporate different levels of structure sparsity regularization that effectively extract discriminative features from different modalities and different GMM components in FV. In order to only consider GMM components in the FV which are discriminative, we first enforce inter-component sparsity to discount unnecessary components. Second, we propose to enhance intra-modal component sparsity with inter-modal non-sparsity. In this way, we encourage discriminative features in different modalities to co-exist. Finally, by learning and combining regressors for both proposal-based FV features and full-image CNN features, our method outperforms state-of-the-art methods on the SUNRGBD Dataset [90] and NYU Depth Dataset V2 [70].

Fig. 5.1 shows an overview of the proposed framework.

5.1 Multi-modal Proposal-based Global Feature Representation

In our framework, local information is incorporated through the use of region proposals and their corresponding local CNN features. More specifically, we use the publicly available proposal extractor [31] to extract region proposals from each RGB-D image. For each proposal, the local CNN features are then computed from both color and geometry data. In addition to the two modalities (RGB and HHA) used in [31], we further include a third modality of *surface normals* (SN) into our framework, represented as unit 3D vectors. Since all three modalities comprise three channels each, we start with

the same 8-layer CNN model pretrained on the Places Dataset [114] for each modality, but then fine-tuned independently. We use the activations of the first fully connected layer (fc6, i.e. layer 6 in the 8-layer CNN) in each modality as the CNN features for each proposal. In order to reduce computational complexity, the number of dimensions d , in the CNN fc6 activation vectors is reduced from 4096 to 400 per modality via PCA. In this way, given an RGB-D image with J extracted object proposals, each proposal in each modality is represented by its corresponding CNN feature vector $\vec{f}_{ij} \in \mathbb{R}^d$.

The CNN features for all proposals within a single RGB-D image is then encoded with the standard Fisher Vector (FV) [74, 79] approach. The FV encoding consists of a K -component Gaussian Mixture Model (GMM) with parameters of $\lambda_k = \{w_k, \vec{\mu}_k, \Sigma_k\}$, $k = 1 \dots K$, where w_k , $\vec{\mu}_k$ and Σ_k are the mixing weight, mean and covariance matrix (assumed diagonal) of the k -th Gaussian component respectively. The gradient vectors (w.r.t. mean $\vec{\mu}_k$ and standard deviation $\vec{\sigma}_k$) are:

$$\begin{aligned} \vec{g}_{\vec{\mu}_k}^i &= \frac{1}{\sqrt{w_k}} \sum_{j=1}^J \gamma_j^i(k) \left(\frac{\vec{f}_{ij} - \vec{\mu}_k}{\vec{\sigma}_k} \right) \\ \vec{g}_{\vec{\sigma}_k}^i &= \frac{1}{\sqrt{2w_k}} \sum_{j=1}^J \gamma_j^i(k) \left(\frac{(\vec{f}_{ij} - \vec{\mu}_k)^2}{\vec{\sigma}_k^2} - 1 \right) \end{aligned} \quad (5.1)$$

where $\gamma_j^i(k)$ is the soft assignment weight of $\vec{f}_{ij}$ to the k -th component:

$$\gamma_j^i(k) = P(k | \vec{f}_{ij}, \lambda). \quad (5.2)$$

Concatenating the two gradient vectors leads to a $2Kd$ -dimensional FV for each modality. By further collating FVs from the three modalities, we obtain a multi-modal feature representation for image i , given by $\vec{x}_i \in \mathbb{R}^D$, where $D = 6Kd$.

5.2 Modality and Component Aware Feature Fusion

5.2.1 Formulation

Let $X = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N] \in \mathbb{R}^{D \times N}$ denote the multi-modal FVs derived from N input RGB-D images, $Y \in \mathbb{R}^{N \times C}$ be the ground truth label matrix with C classes, and $W \in \mathbb{R}^{D \times C}$ be the transformation or weight matrix that maps input features X into the label domain via $X^T W$.

We formulate our method as solving a regression problem with several regularization terms:

$$\begin{aligned} \min_W F &= R + R_1 + R_2 + R_3 \\ &= \frac{1}{2} \|X^T W - Y\|_F^2 \\ &\quad + \lambda_1 \|W_{(P)}\|_2^1 + \lambda_2 \|W_{(Q)}\|_1^2 + \lambda_3 \|W\|_1 \end{aligned} \tag{5.3}$$

The first term R is the standard least-squares regression term. It encourages the transformation $X^T W$ to closely reconstruct the labels, biasing towards a W that extracts discriminative information from the features. R_3 is the common l_1 -norm regularization term to invoke only a sparse set of features, while R_1 and R_2 are explained in detail below. λ_1 , λ_2 and λ_3 are positive tradeoff parameters.

Component Regularization Term R_1 : Since the GMM components of the FV encoding are constructed from all region proposals, which are obtained in a generic fashion, many of these components do not contribute discriminative power for distinguishing between scene classes. Thus, we propose a regularization term based on group lasso [108] which should result in only the expected few discriminative components being associated with large weights, while the remaining components will be associated with zero or small

Figure 5.2: Illustration of the component regularization term, where each component is treated as a group and each group is encouraged to have either all zero weights (white squares) or multiple nonzero weights (colorful squares).

weights. Specifically we define

$$R_1 = \|W_{(P)}\|_2^1 = \sum_{j=1}^C \sum_{l=1}^L \left\| (W_{(P)})_l^j \right\|_2 \quad (5.4)$$

where $(W_{(P)})_l^j \in \mathbb{R}^{2d}$ denotes the weights for the l -th component of the j -th class. There are K components for each modality, resulting in $L = M \times K$ components in total, where M is the number of modalities ($M = 3$). Eq. (5.4) essentially applies l_2 -norm regularization within each component (because the parameters of a component should have similar importance) and l_1 -norm regularization across different components. Fig. 5.2 illustrates the idea of the component-based regularization, where a component is encouraged to have either all zero weights or multiple non-zero weights.

Modality Regularization Term R_2 : Although it may be that the discriminative power of different modalities are different, it is expected that for the sparse set of discriminative features, their discriminability comes from a mixture of modalities, rather than due to a single modality in isolation (i.e. scene classification will not be optimally performed using only data from one modality). Thus, we propose to use the regularization term of

Figure 5.3: Illustration of the modality regularization term, where each modality is treated as a group and each group is encouraged to have sparse nonzero weights (colorful squares) and many zero weights (white squares).

exclusive group lasso [115] to encourage discriminative features from different modalities to co-exist, while features within one modality are encouraged to compete with each other. Fig. 5.3 illustrates the idea of modality regularization, where each modality is encouraged to be associated with sparse non-zero weights within itself, but not so across different modalities. We define the modality regularization term as:

$$R_2 = \|W_{(Q)}\|_1^2 = \sum_{j=1}^C \sum_{m=1}^M (\|(W_{(Q)})_m^j\|_1)^2 \quad (5.5)$$

where $(W_{(Q)})_i^j \in \mathbb{R}^{2Kd}$ denotes the weights for the i -th modality of the j -th class. Eq. (5.5) essentially applies l_1 -norm regularization within each modality to encourage sparsity and l_2 -norm like regularization across different modalities to encourage balance.

5.2.2 Optimization

To optimize the transformation matrix W in (5.3), we compute the derivative of the overall cost function w.r.t. $W_j \in \mathbb{R}^D$ for class j , based on existing solutions developed

for the lasso, group lasso and exclusive lasso techniques. Specifically:

$$\begin{aligned} \frac{\partial F}{\partial W_j} &= XX^T W_j - X\vec{y}_j \\ &+ \lambda_1 D_j^{(1)} W_j + 2\lambda_2 D_j^{(2)} W_j + \lambda_3 D_j^{(3)} W_j \end{aligned} \quad (5.6)$$

where $\vec{y}_j \in \mathbb{R}^N$ denotes the label vector for all training images in class j , while $D_j^{(1)}$, $D_j^{(2)}$ and $D_j^{(3)}$ are all diagonal $D \times D$ matrices dependent on W_j . The i -th diagonal elements of $D_j^{(1)}$, $D_j^{(2)}$ and $D_j^{(3)}$ are calculated as

$$\begin{aligned} D_{ij}^{(1)} &= \frac{1}{\|(W_{(P)})_i^j\|_2} \\ D_{ij}^{(2)} &= \frac{\|(W_{(Q)})_m^j\|_1}{|W_{ij}|} \\ D_{ij}^{(3)} &= \frac{1}{2|W_{ij}|} \end{aligned} \quad (5.7)$$

Detailed derivations can be found in [47, 95, 96].

Once the derivative $\frac{\partial F}{\partial W_j}$ is available, W_j is updated as

$$W_j \leftarrow W_j - \gamma \frac{\partial F}{\partial W_j} \quad (5.8)$$

where γ is the learning rate. As $D_j^{(1)}$, $D_j^{(2)}$ and $D_j^{(3)}$ depend on W_j , we update D_j and W_j in an iterative way. The optimization pipeline is shown in Algorithm 3.

Using this optimization procedure, we learn an optimal transformation matrix W . In the testing stage, once the multi-modal FV features X of a test RGB-D image have been extracted, the regression values are computed simply using $X^T W$, with the maximum regression value regarded as the classification result.

To further leverage global features, we also adapt the proposed feature fusion framework to the multi-modal CNN features applied on full images. Compared with the proposal-based feature fusion framework, the only difference is that the full-image based framework does not have components, because it is a single measurement rather than modeled

Algorithm 3: The optimization pipeline

Input: X : multi-modal FV features;

Y : ground-truth label matrix.

Output: W : transformation matrix.

Step 1 (Initialization):

Initialize W as zero matrix.

Step 2 (Optimization):

For each class j

While not converged do

2.1. Fixing W_j , update $D_j^{(1)}$,
 $D_j^{(2)}$ and $D_j^{(3)}$ according to (5.7).

2.2. Fixing $D_j^{(1)}$, $D_j^{(2)}$ and $D_j^{(3)}$,
update W_j according to (5.8).

end while until convergence

end for

as a distribution. In other words, the cost function of the full-image based framework only contains R , R_2 and R_3 terms of (5.3). Finally, the regression values from both the proposal-based and the full-image based frameworks are added to obtain the final classification.

5.3 Experiments

To evaluate the effectiveness of our proposed modality and component aware feature fusion framework, we perform scene classification experiments on the SUNRGBD Dataset [90] and the NYU Depth Dataset V2 [86]. The details of the experiments and the results are described in the following subsections.

Figure 5.4: Confusion matrices of ‘FV (L1)’ (left) and ‘FV (Modality+Component+L1)’ (right) on SUNRGBD Dataset. It shows that by adding the modality and component regularization terms, the performance is improved for almost all the classes.

5.3.1 Datasets and Experimental Setup

SUNRGBD Dataset: This dataset has 19 scene categories. It consists of 10,335 RGB-D scene images, including 3,784 Kinect v2 images, 1,159 Intel RealSense images captured by Song *et al.* [90], 1,449 Kinect v1 images taken from the NYU Depth Dataset V2 [86], 554 Kinect v1 images selected from the Berkeley B3DO Dataset [38], and 3,389 Asus Xtion images selected from SUN3D videos [100]. We follow the experimental settings stated in [90] and only keep categories with more than 80 images. Using the publicly available split, there are in total 4,845 images for training and 4,659 images for testing.

NYU Depth Dataset V2: This dataset consists of 1,449 images. It has 27 scene categories but only a few of them are well represented. Following the procedures stated in [30], the original 27 categories are reorganized into 10 scene categories, including the 9 most common categories and an ‘other’ category for images in the remaining cat-

egories. We use the publicly available split, which has 795 images for training and 654 images for testing.

Metrics: For both datasets, we report the means of diagonal values of the confusion matrices, which are the average precisions over all scene classes. Another metric we considered is the overall accuracy, which is the precision over all test images. Since we found these two metrics to be strongly correlated, only the former is listed for presentation conciseness.

Fine-tuning: Our starting point is the current state-of-the-art CNN model (Places-CNN) for scene classification, pre-trained on the Places Dataset [114] (2.5 million RGB images with 205 scene categories). To better adapt the pre-trained CNN network for RGB-D data, especially for the HHA and surface normal modalities, we fine-tuned the Places-CNN with our relevant data. For the SUNRGBD Dataset, we fine-tuned the Places-CNN with each of the three modalities (RGB, HHA and surface normals) from training images, utilizing image-level labels. For the NYU Depth Dataset V2, the fine-tuning was carried out in two stages: first with images from the SUNRGBD Dataset (but excluding the NYU V2 images), then using training images from the NYU Depth Dataset V2.

After one of the fine-tuned CNNs has operated on a region proposal in an image, the CNN activation vector of the first fully connected layer (fc6) is extracted. As stated previously, based on a collection of such vectors in training images, PCA is then used to reduce these 4096-dimensional vectors to 400 dimensions, and further encoded as GMM-based Fisher vectors.

Parameters: The parameter settings for the two datasets are identical. The number of GMM components K for each modality is 64. The PCA-reduced dimensionality of the CNN activation vector is 400. The parameters λ_1 , λ_2 and λ_3 in (5.3) for proposal-based feature fusion are set at 0.005, 0.01 and 0.001 respectively. For full-image-based feature

Table 5.1: Comparing the classification results of the proposal based FV features and the full-image based CNN features under different modalities with linear SVM classifier on SUNRGBD Dataset.

Accuracy (%)	Full (SVM)	FV (SVM)	FV+Full (SVM)
RGB	40.4	36.2	-
HHA	36.3	34.6	-
SN	34.3	30.6	-
RGB+HHA	44.9	39.7	-
RGB+HHA+SN	45.7	41.2	45.9

Table 5.2: Comparison of different baselines of our proposed feature fusion framework on SUNRGBD Dataset.

Method	Accuracy (%)
FV (SVM)	41.2
FV (L1)	41.0
FV (Modality + L1)	43.9
FV (Component + L1)	42.7
FV (Modality+Component+L1)	45.1
Full (SVM)	45.7
Full (L1)	44.9
Full (Modality + L1)	45.4
Combine FV and Full	48.1

fusion ($\lambda_1 = 0$), we set λ_2 and λ_3 to be 0.001 and 0.0001 respectively. The learning rates γ in (5.8) are set at 10^{-4} and 10^{-8} for proposal-based and full-image-based feature fusion respectively. When optimizing W_j for each class, the number of iterations is fixed at 100.

5.3.2 Results on SUNRGBD Dataset

We first compare the linear SVM classification results of the proposal-based FV features and the full-image-based CNN features obtained from different combinations of

Table 5.3: Comparison with state-of-the-art methods on SUNRGBD Dataset.

Method	Accuracy (%)
Song <i>et al.</i> [90]	39.0
Liao <i>et al.</i> [60]	41.3
MCAFF	48.1

modalities, without including our proposed regularization terms. We considered three baselines: 1) ‘Full (SVM)’: the full-image-based CNN features with SVM; 2) ‘FV (SVM)’: the proposal-based FV features with SVM; and 3) ‘FV+Full (SVM)’: concatenating the full-image features and the FV features prior to linear SVM classification. Table 5.1 shows the comparison results. Among the three individual modalities, RGB features achieve the best performance; however, it is clear that combination of the three modalities substantially improves performance. The comparisons between ‘RGB+HHA’ and ‘RGB+HHA+SN’ indicate that expressing surface normals (SN) as explicitly separate from HHA leads to improved performance, although both are indirectly extracted from depth images. More importantly, we can see that ‘FV (SVM)’ performs poorly compared with ‘Full (SVM)’, despite ‘FV (SVM)’ features having $51,200$ dimensions ($D = 6Kd = 6 \times 64 \times 400$) while ‘Full (SVM)’ features only have 4096×3 dimensions. Even the combination of ‘FV+Full (SVM)’ only slightly improves the performance. This is mainly due to many dimensions of the FV features not having discriminative power but which cause regressor overfitting, unless better regularization is used (as implemented in our proposed feature fusion framework).

Table 5.2 shows the impact of our modality and component aware feature fusion frameworks with the added regularization terms. Here we consider seven other settings: 1) ‘FV (L1)’: using our framework only on proposal-based CNN features and with only the R_3 (L1-norm) regularization term active; 2) ‘FV (Modality+L1)’: proposal-based fea-

tures only with R_2 and R_3 active; 3) ‘FV (Component+L1)’: proposal-based features only with R_1 and R_3 active; 4) ‘FV (Modality+Component+L1)’: proposal-based features only with R_1 , R_2 and R_3 active; 5) ‘Full (L1)’: using our framework only on full-image based CNN features, with only R_3 active; 6) ‘Full (Modality+L1)’: full-image features only with R_2 and R_3 active; 7) ‘Combine FV and Full’: combined regression using both ‘FV (Modality+Component+L1)’ and ‘Full (Modality+L1)’, which is our final result.

From Table 5.2, we can see that our feature fusion framework is very effective for the multi-modal FV features, greatly improving the performance from 41.0% under the setting of ‘FV (L1)’ to 45.1% under the setting of ‘FV (Modality+Component+L1)’. It demonstrates that the discriminative information of the high-dimensional multi-modal FV features can be better extracted with the developed structure sparsity regularization. Although ‘Full (Modality+L1)’ does not outperform ‘Full (SVM)’ (mainly because the full-image based CNN features are not of high dimensions), the combination of the regression results of the FV features and the full-image based features (‘Combine FV and Full’) achieves the best performance. This suggests that the proposal-based features contain pertinent local information not represented in full-image-based features.

Table 5.3 shows comparison with state-of-the-art methods. Our method is called MCAF-F (Modality and Component Aware Feature Fusion). We compared with: 1) Song *et al.* [90], which directly uses pre-trained Places-CNN to extract features from RGB and HHA followed by RBF kernel SVM for classification; and 2) Liao *et al.* [60], which incorporates features extracted from semantic segmentation to improve scene classification. It can be seen that our proposed method significantly outperforms the two state-of-the-art methods.

In Fig. 5.4, we visualize the confusion matrix to give the performance comparison between ‘FV (L1)’ and ‘FV (Modality+Component+L1)’. It can be seen that there is a performance improvement for almost every class. We can also spot some misclassification

Figure 5.5: The effect of choosing different λ_2 .

cases, *e.g.* many ‘lab’ images are misclassified as ‘office’, and some ‘lecture_theatre’ images are misclassified as ‘classroom’. These are due to both visual and semantic similarity between such classes. Fig. 5.5 shows the effect of choosing different λ_2 when fixing λ_1 and λ_3 as 0.005 and 0.001.

5.3.3 Results on NYU Depth Dataset V2

We also obtained results on NYU Depth Dataset V2, where we can make similar observations to those for the SUNRGBD Dataset. Table 5.4 compares the classification results of the proposal-based FV features and the full-image-based CNN features with linear SVM classifier. Table 5.5 compares the results under different baseline settings of our modality and component aware feature fusion framework. In this dataset, we can also see that the ‘FV (Modality+Component+L1)’ baseline significantly outperforms ‘FV (L1)’ by over 6%, which further proves the effectiveness of the structured sparsity promoted by our feature fusion method. By combining the regression results of ‘FV

Table 5.4: Comparing the classification results of the proposal based FV features and the full-image based CNN features under different modalities with linear SVM classifier on NYU Depth Dataset V2.

Accuracy (%)	Full (SVM)	FV (SVM)	FV+Full (SVM)
RGB	53.5	49.2	-
HHA	51.5	52.2	-
SN	51.7	44.8	-
RGB+HHA+SN	58.5	55.8	58.7

Table 5.5: Comparison of different baselines of our proposed feature fusion framework on NYU Depth Dataset V2.

Method	Accuracy (%)
FV (SVM)	55.8
FV (L1)	53.5
FV (Modality + L1)	56.7
FV (Component + L1)	55.5
FV (Modality+Component+L1)	59.8
Full (SVM)	58.5
Full (L1)	58.8
Full (Modality + L1)	59.1
Combine FV and Full	63.9

(Modality+Component+L1)’ and ’Full (Modality+L1)’, we further improve the performance significantly.

Table 5.6 shows comparisons with state-of-the-art methods. Gupta *et al.* [29,30] used the semantic segmentation output (i.e. the probabilities of belonging to different semantic classes) as local features and applied spatial pyramid (SPM) on them. We also show the results of the three baselines in [29]: 1) histograms of vector quantized color SIFT as features with SPM; 2) histograms of geocentric textons with SPM; 3) combination of 1) and 2) with SPM.

Figure 5.6: Confusion matrix of our final results ('Combine FV and Full') on NYU Depth Dataset V2.

Table 5.6: Comparison with state-of-the-art methods on NYU Depth Dataset V2. We reimplemented the second-order pooling method [5] and show the results as 'O2P'.

Method	Accuracy (%)
Gupta <i>et al.</i> [29]	45.4
SPM on SIFT [29]	38.9
SPM on G. Textons [29]	33.8
SPM on SIFT+G. Textons [29]	44.9
O2P on color SIFT and LBP	41.0
O2P on depth SIFT and LBP	48.5
O2P on color+depth	50.9
MCAFF	63.9

Recently, Banica *et al.* [5] made use of second-order pooling (O2P) [13] of hand-crafted features mainly for the RGB-D semantic segmentation problem, but they also directly apply O2P features for scene classification as an additional application. For RGB-D scene classification on NYU Depth Dataset V2, they reported a very high classification results of 83.81%. Despite our careful reimplementation of their method in detailed consultation with one of the authors, we were unable to reproduce and verify their published percentages; hence we only list the results obtained from our implementation of the O2P method in Table 5.6*. Specifically, we conduct second-order pooling on SIFT and Local Binary Patterns (LBP) for both color and depth images. The pooling was done in subregions of a $1, 2 \times 2$ and 4×4 SPM. Fig. 5.6 shows the confusion matrix of our final results ('Combine FV and Full'). We can see that the results of 'home_office' and 'other' classes were not as good as other classes, since 'others' is not well defined, while 'home_office' is significantly confused with 'living_room'.

5.4 Summary

In this chapter, we describe our method for RGB-D scene classification task. We propose a modality and component aware feature fusion framework, where we fuse Fisher Vector features from different modalities. In particular, we apply group lasso to GM-M components and exclusive group lasso across modalities to enforce both component sparsity and modal non-sparsity. In this way, discriminative feature dimensions are associated with larger weights. We conduct experiments on SUNRGBD dataset and NYU depth dataset V2. Our method outperforms the state-of-the-art methods.

*We attempted a careful reimplementation in consultation with an author of [5] based on codes of [13], who was also unable to figure out the reason for the discrepancy in results.

Chapter 6

Conclusions and Future Work

6.1 Conclusion

In this thesis, we have studied related topics of feature learning for RGB-D scene understanding and proposed solutions for scene labeling, object recognition and scene classification, respectively.

Firstly, in Chapter 3, we have presented an unsupervised feature learning framework that learns features from RGB-D data for scene labeling task. Our method considers unsupervised feature learning and feature encoding problems together and implicitly infers the relationship between two modalities. Furthermore, we have also extended the framework with stacked nonlinear projections to make it more general. By stacking the learning framework, our method could learn hierarchical features. Linear SVMs are trained on superpixels to produce the final labeling. We conducted experiments on NYU depth dataset V1 and V2 and obtained comparable results with state-of-the-art methods including those using hand-crafted features and those learned features from raw data.

Secondly, in Chapter 4, we have proposed a CNN-based multi-modal feature learning framework for RGB-D object recognition task. Instead of fusing color and depth data from the outset or concatenating separately learned features before the classification,

we extract both features with shared common patterns and features with modal-specific patterns in a joint framework. The experimental results show that our method integrated with CNN layers greatly boosts the performance. Our method outperforms state-of-the-art approaches on both of RGB-D Object Dataset and 2D3D Dataset.

Thirdly, in Chapter 5, we have proposed a modality and component aware feature fusion framework that effectively makes use of high-dimensional FV features from RGB, HHA and surface normal modalities. We formulate our method as a regression problem with regularization terms corresponding to modality and component related structure sparsity. By combining the regression results of the proposal based multi-modal FV features and the full-image based multi-modal CNN features, our method outperforms state-of-the-art methods for scene classification on the SUNRGBD Dataset and the NYU Depth Dataset V2.

6.2 Future Work

This thesis focuses on RGB-D scene understanding and feature learning problems. Four directions can be explored for future works:

- **Scene labeling with contextual information:** Currently, our proposed method for scene labeling learns feature of each patch independently, which ignores the contextual information in other image regions. One future direction could be feature learning considering contextual information for scene labeling. This concept is different from that of the traditional methods, which conduct an MRF or CRF process after the feature extraction process. The feature learning process itself should consider the contextual information. In this way, the learned features could contain the contextual knowledge and be more discriminative.

- **Multi-modal problem for other tasks:** In this thesis, we have discussed some related methods to deal with multi-modal problems. Currently, we focus on RGB-D scene understanding. In the future, we could make use of multi-modal fusion methods on other tasks. For example, there are a large amount of images on social media. Many images are with captions or short describing texts. Image and text could be fused together as two modalities. Carefully designed fusion methods could boost the performance than direct concatenation.
- **Holistic scene understanding:** In this thesis, the tasks of scene understanding including scene labeling, object recognition, scene classification are tackled separately. Each isolated task suffers from information limitation, e.g. the scene labeling task needs help provided by object recognition to indicate higher-level features instead of just local information, the object detection would be easier if the scene type information is available.

Thus, another interesting direction of future work is holistic scene understanding combining tasks of scene labeling, object detection and scene classification. Rather than treating other tasks as black boxes, all tasks could be combined with joint optimization. By jointly inferring these tasks, high-level and low-level clues could be guided by each other and errors could be reduced. In this way, the understanding of the indoor scene could be fine-grained with higher accuracy.

- **Dynamic scene understanding:** In robotics applications, obtaining semantic labels for image sequences in RGB-D videos is significant for robots to interact with the dynamic environment. By quickly identifying objects when the robot is navigating the room, robots could decide where to move next and how to manipulate with objects. Or when there are people nearby, robots could decide the next actions to help people by understanding the dynamic change. The labels should be

consistent across the video. By combining temporal-spatial information, the system would be more robust and efficient. To achieve that, a fast system is needed. By distributing the scene understanding structure on current GPUs with powerful computing capacity, we expect to work out a system with high speed.

References

- [1] A. E. Abdel-Hakim and A. A. Farag. Csift: A sift descriptor with color invariant characteristics. In *CVPR*, pages 1978–1983, 2006. 12
- [2] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *TSP*, 54(11):4311–4322, 2006. 12
- [3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, pages 898–916, 2011. 28
- [4] M. Avriel. *Nonlinear programming: Analysis and methods*. Prentice-Hall, 1976. 32
- [5] D. Banica and C. Sminchisescu. Second-order constrained parametric proposals and sequential search-based structured prediction for semantic segmentation in rgb-d images. In *CVPR*, pages 3517–3526, 2015. xvi, 6, 14, 87, 88
- [6] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417. 2006. 12
- [7] M. Blum, J. T. Springenberg, J. Wulfin, and M. Riedmiller. A learned feature descriptor for object recognition in rgb-d data. In *ICRA*, pages 1298–1303, 2012. 5, 12, 13, 63, 64
- [8] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *IROS*, pages 821–826, 2011. 12
- [9] L. Bo, X. Ren, and D. Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *NIPS*, pages 2115–2123, 2011. 5, 13
- [10] L. Bo, X. Ren, and D. Fox. Unsupervised Feature Learning for RGB-D Based Object Recognition. In *ISER*, pages 387–402, 2012. 5, 12, 13, 59, 61, 64, 65, 66
- [11] B. Browatzki, J. Fischer, B. Graf, H. Bulthoff, and C. Wallraven. Going into depth: Evaluating 2d and 3d cues for object classification on a new, large-scale object dataset. In *ICCV workshop*, pages 1189–1195, 2011. 58, 59, 65, 66
- [12] C. Cadena and J. Košecka. Semantic parsing for priming object detection in rgb-d scenes. In *Workshop on Semantic Perception, Mapping and Exploration*, 2013. 4, 10, 40, 41, 42, 44

REFERENCES

- [13] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, pages 430–443. 2012. 14, 88
- [14] A. Coates, A. Y. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *International Conference on Artificial Intelligence and Statistics*, pages 215–223, 2011. 22
- [15] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013. 4, 11, 40, 41, 42, 44, 51
- [16] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 4, 9
- [17] H. Daumé III. Frustratingly easy domain adaptation. *ACL*, page 256, 2007. 16
- [18] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei. Imagenet large scale visual recognition competition 2012 (ilsvrc2012), 2012. 15
- [19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5, 13, 15
- [20] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, pages 184–199. 2014. 15
- [21] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *arXiv preprint arXiv:1411.4734*, 2014. 11, 41
- [22] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *PAMI*, 35(8):1915–1929, 2013. 3, 9, 11, 15
- [23] S. Foix, G. Alenya, and C. Torras. Lock-in time-of-flight (tof) cameras: a survey. *Sensors Journal, IEEE*, 11(9):1917–1926, 2011. 2
- [24] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *CVPR*, pages 1–8, 2008. 3, 9
- [25] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587. 2014. 15
- [26] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, pages 392–407. 2014. 6, 13
- [27] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 10
- [28] D. Grangier, L. Bottou, and R. Collobert. Deep convolutional networks for scene parsing. In *ICML Deep Learning Workshop*, 2009. 3, 9

REFERENCES

- [29] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation. *IJCV*, 112(2):133–149, 2014. 14, 86, 87
- [30] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, pages 564–571, 2013. 4, 6, 10, 14, 80, 86
- [31] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, pages 345–360. 2014. 71, 73
- [32] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *ECCV*, pages 216–229. 2012. 29
- [33] X. He, R. Zemel, and M. Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 9
- [34] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *ICRA*, pages 2631–2638, 2014. 11
- [35] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006. 15
- [36] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *CVPR*, pages 5344–5352, 2015. 16
- [37] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. *JMLR*, 12:3371–3412, 2011. 17
- [38] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. Springer, 2013. 80
- [39] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010. 13
- [40] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013. 15
- [41] Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. In *NIPS*, pages 982–990, 2010. 16
- [42] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *PAMI*, 21(5):433–449, 1999. 12
- [43] L. Kang, P. Ye, Y. Li, and D. Doermann. Convolutional neural networks for no-reference image quality assessment. In *CVPR*, pages 1733–1740, 2014. 15
- [44] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014. 15

REFERENCES

- [45] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri. Geometry driven semantic labeling of indoor scenes. In *ECCV*, pages 679–694, 2014. 4, 10, 35, 41, 42, 44
- [46] A. Klami and S. Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72(1):39–46, 2008. 16
- [47] D. Kong, R. Fujimaki, J. Liu, F. Nie, and C. Ding. Exclusive feature learning on arbitrary structures via $l_{1,2}$ -norm. In *NIPS*, pages 1655–1663, 2014. 17, 78
- [48] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, pages 244–252, 2011. 5, 10
- [49] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 5, 13, 15, 59, 60
- [50] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *ICRA*, pages 3050–3057, 2014. 11
- [51] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *ICRA*, pages 1817–1824, 2011. 12, 58, 63, 64
- [52] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In *NIPS*, pages 1017–1025, 2011. 4, 22, 54
- [53] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, pages 3361–3368, 2011. 4, 22
- [54] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 15
- [55] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808, 2006. 25
- [56] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, pages 609–616, 2009. 15, 26
- [57] V. S. Lempitsky, A. Vedaldi, and A. Zisserman. Pylon model for semantic segmentation. In *NIPS*, pages 1485–1493, 2011. 3, 9
- [58] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *IJRR*, 34(4-5):705–724, 2015. 16
- [59] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *IJCV*, 43(1):29–44, 2001. 12
- [60] Y. Liao, S. Kodagoda, Y. Wang, L. Shi, and Y. Liu. Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks. *arXiv preprint arXiv:1509.06470*, 2015. 6, 14, 83, 84

REFERENCES

- [61] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 2011. 10
- [62] J. Liu, Y. Jiang, Z. Li, Z.-H. Zhou, and H. Lu. Partially shared latent factor learning with multiview data. *Neural Networks and Learning Systems, IEEE Transactions on*, 2014. 16
- [63] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999. 3, 9
- [64] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 12
- [65] Microsoft Kinect. <http://www.xbox.com/en-us/kinect>. 2
- [66] B. Micusik and J. Kosecka. Semantic segmentation of street scenes by superpixel co-occurrence and 3d geometry. In *ICCV Workshops*, pages 625–632, 2009. 3, 9
- [67] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013. 15
- [68] A. Muller and S. Behnke. Learning depth-sensitive conditional random fields for semantic segmentation of rgb-d images. In *ICRA*, pages 6232–6237, 2014. 10
- [69] N. Naikal, A. Y. Yang, and S. S. Sastry. Informative feature selection for object recognition via sparse pca. In *ICCV*, pages 818–825, 2011. 17
- [70] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. 73
- [71] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint ℓ_2 , ℓ_1 -norms minimization. In *NIPS*, pages 1813–1821, 2010. 57
- [72] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *PAMI*, 24(7):971–987, 2002. 3, 4
- [73] D. Pei, H. Liu, Y. Liu, and F. Sun. Unsupervised multimodal feature learning for semantic image segmentation. In *IJCNN*, 2013. 4, 11, 35, 38
- [74] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156. 2010. 6, 74
- [75] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, pages 413–420, 2009. 1
- [76] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, pages 2759–2766, 2012. 4, 10, 28, 34, 35, 38, 39
- [77] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *CoRR*, abs/1409.0575, 2014. 63

REFERENCES

- [78] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *International Conference on Artificial Intelligence and Statistics*, pages 448–455, 2009. 15
- [79] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013. 6, 74
- [80] M. Schmidt. minfunc. 2005. 25
- [81] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. In *ICPR*, pages 3288–3291, 2012. 15
- [82] S. Shirazi, M. T. Harandi, B. C. Lovell, and C. Sanderson. Object tracking via non-euclidean geometry: A grassmann approach. In *WACV*, pages 901–908, 2014. 29
- [83] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*. 2006. 9
- [84] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 601–608. IEEE, 2011. 4, 5, 33, 34, 35, 38
- [85] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *ICCV Workshops*, pages 601–608, 2011. 10
- [86] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, pages 746–760, 2012. 33, 34, 79, 80
- [87] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013. 15
- [88] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3d object classification. In *NIPS*, 2012. 5, 12, 13, 64
- [89] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, pages 129–136, 2011. 3, 9
- [90] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, pages 567–576, 2015. 6, 14, 73, 79, 80, 83, 84
- [91] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014. 15
- [92] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 17
- [93] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014. 15

REFERENCES

- [94] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, 2008. 15
- [95] H. Wang, F. Nie, H. Huang, S. Risacher, C. Ding, A. J. Saykin, and L. Shen. Sparse multi-task regression and feature selection to identify brain imaging predictors for memory performance. In *ICCV*, pages 557–562, 2011. 78
- [96] H. Wang, F. Nie, H. Huang, S. L. Risacher, A. J. Saykin, L. Shen, et al. Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via sparse multimodal multitask learning. *Bioinformatics*, 28(12):i127–i136, 2012. 78
- [97] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, pages 124.1–124.11, 2009. 4, 22
- [98] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, 2010. 5
- [99] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *PAMI*, 31(2):210–227, 2009. 17
- [100] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, pages 1625–1632, 2013. 80
- [101] Y. Xie, J. Ho, and B. Vemuri. On a nonlinear generalization of sparse coding and dictionary learning. In *ICML*, page 1480, 2013. 29
- [102] H. Yang, J. T. Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai. Can partial strong labels boost multi-label object recognition? *arXiv preprint arXiv:1504.05843*, 2015. 14
- [103] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *TIP*, 19(11):2861–2873, 2010. 17
- [104] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801, 2009. 25
- [105] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, pages 1331–1338, 2011. 17
- [106] D. Yoo, S. Park, J.-Y. Lee, and I. S. Kweon. Fisher kernel for deep neural activations. *arXiv preprint arXiv:1412.1628*, 2014. 6, 14
- [107] K. Yu, Y. Lin, and J. Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. In *CVPR*, pages 1713–1720, 2011. 5
- [108] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. 17, 75

REFERENCES

- [109] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. 2014. 15
- [110] X. Zeng, W. Ouyang, M. Wang, and X. Wang. Deep learning of scene-specific classifier for pedestrian detection. In *ECCV*, pages 472–487, 2014. 15
- [111] Q. Zhang, G. Hua, W. Liu, Z. Liu, and Z. Zhang. Can visual recognition benefit from auxiliary information in training? In *ACCV*, 2014. 17, 64
- [112] X. Zhang and Y. LeCun. Text understanding from scratch. *arXiv preprint arXiv:1502.01710*, 2015. 15
- [113] Y. Zhang, X.-s. Wei, J. Wu, J. Cai, J. Lu, V.-A. Nguyen, and M. N. Do. Weakly supervised fine-grained image categorization. *arXiv preprint arXiv:1504.04943*, 2015. 14
- [114] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014. 5, 13, 14, 74, 81
- [115] Y. Zhou, R. Jin, and S. Hoi. Exclusive lasso for multi-task feature selection. In *International Conference on Artificial Intelligence and Statistics*, pages 988–995, 2010. 17, 18, 77
- [116] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiang. Learning discriminative and shareable features for scene classification. In *ECCV*, pages 552–568. 2014. 6, 14