# Deep Learning for Visual Understanding

**Proefschrift**

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op donderdag 5 oktober 2017
klokke 10.00 uur

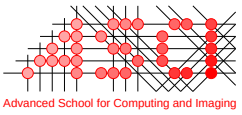
door

**Yanming Guo**

geboren te Hebei, China
in 1989

**Promotiecommissie**

| | |
|---|---|
| Promotor: | Prof. Dr. J.N. Kok |
| Co-promotor: | Dr. M.S. Lew |
| Overige leden: | Prof. Dr. A. Plaat |
| | Prof. Dr. W. Kraaij |
| | Prof. Dr. T.H.W. Bäck |
| | Prof. Dr. H. Trautmann (University of Münster, Germany) |
| | Prof. Dr. S. Rüger (Open University, United Kingdom) |




This work was carried out in the ASCI graduate school.
ASCI dissertation series number: 378



ISBN: 978-94-6299-729-5

Printed by: Ridderprint, the Netherlands

This research is financially supported by the China Scholarship Council (CSC), Grant No. 201306110026.

# Contents

# Chapter 1

# Introduction

## 1.1 Background

It is now the Age of Information. Each day, users produce enormous amounts of data by modern commonplace technologies. As a result, the amount of information is increasing exponentially, especially the multimedia information. Most of this information is stored digitally and available to the public. For example, it is reported that the Facebook users have uploaded over 250 billion photos, and are uploading 350 million new photos each day. Such a large amount of available data is a double-edged sword in our lives. On the one hand, if we can properly handle and analyze the data, we can have more alternatives for our queries. On the other hand, it is easy for us to get lost in the sea of data. Without computer-aided programs, it may take centuries for us to sift through the information and find what we want.

While search engines like Google and Yahoo can perform the textual analysis quite well, it is still challenging to fully exploit the visual content due to the well-known semantic gap. The key to bridging this gap is to develop or learn highly discriminative features to represent the images.

Generally, the development of image representation can be divided into three stages: In the first stage, images are described with low-level global features,

such as color histograms, contour representations, shape descriptors, and texture features. These features represent the whole image with a single vector, and can capture the global image appearance well. However, they are sensitive to occlusion and clutter.

To incorporate more local context and obtain a more informative description, various local features are developed, such as SIFT [1], SURF [2], HoG [3], etc. The local features are descriptors of local image neighborhoods computed at multiple key points. Compared with global features, local features are more robust to image translation and occlusion. To aggregate spatial local features into a global image representation, these features are often encoded with Bag of Visual Words (BoW) [4], or its variants, such as VLAD [5] or Fisher Vector(FV) [6].

Both the traditional global features and local features are hand-crafted features, which often require expensive human labor and do not generalize well. Recent studies have shown that there are no universally best hand-crafted features for all datasets, and it would be more advantageous to learn features directly from the raw data [7]. Since 2006, deep learning has emerged as a new area of machine learning research [8], and it introduces the concept of end-to-end learning which means transformation from pixel level to real application. Deep learning algorithms typically attempt to distill high-level abstractions in data by utilizing hierarchical architectures. The output of each intermediate layer can be viewed as a representation of the original input data. Several deep representations have been repeatedly verified to be highly discriminative and achieved top tier performance on various benchmark datasets and international contests.

Due to advances in deep image representations, numerous breakthroughs have been made in diverse computer vision applications. For example, for the most intuitive and extensively studied task, image classification, many deep learning algorithms have reached comparable performance relative to the human performance on the large scale ImageNet dataset [9]. Aside from solely discovering the objects, there are some new emerging applications (e.g. image captioning, visual question answering) which aim to exploit more information (e.g. action, relation and etc) based on the deep image representation, and also achieved competitive results with the human performance [10].

## 1.2 Research Goals and Contributions

The main purpose of our work is to develop new algorithms which can improve the understanding of images. To fulfill this, we focus on two visual applications: image classification and image captioning.

Image classification aims to classify images into pre-defined categories, and helps people to know what objects the images contain. In the first part of this thesis, we propose new features which can improve the performance without significantly increasing the computational cost. Therefore, they may be utilized in many other applications.

The second part of the thesis proposes to address the hierarchical image classification task, which can generate multiple hierarchical labels in a coarse-to-fine pattern. By providing the evolution of the image categories, this task can better describe what the categories are, especially for the fine-grained categories.

For the third part, we investigate a more challenging and new emerging task, i.e. image captioning, which attempts to generate a sentence to describe the image. In contrast to image classification which only detects the existence of an object, the sentence generated by image captioning may also contain the action, relation and etc.

## 1.3 Thesis Overview

This thesis is based on articles where I have been a primary author that have been published or are currently under consideration at respected journals and conference proceedings. The following provides a brief description of each chapter.

Chapter 2 presents a survey which reviews about 200 papers published between 2010 and 2016 in the area of deep learning for visual understanding. The survey provides a comprehensive background for this research area, including the

development of the relevant methods, the applications, and the directions that the field is moving towards. This survey has been published by :

- Neurocomputing (journal)

Chapter 3 introduces an effective and straightforward feature, called Principal Pyramidal Convolution (PPC). This feature is derived from the commonly-used CNN feature (i.e. the fully-connected activation), and demonstrates superior performance than the baseline for different datasets and different dimensions. This work has been published in the conference proceeding:

- 16th Pacific-Rim Conference on Multimedia (PCM2015) in Gwangju, Korea.

Chapter 4 presents a new feature called Bag of Surrogate Parts (BoSP). This feature is motivated by the well-known Bag of Words (BoW) scheme, and aims to integrate the advantages of CNN and BoW (i.e. high discrimination for CNN, and scale/position/occlusion invariance for BoW). Together with the feature, several enhancements are also proposed, including spatial aggregation, scale pooling and global-part prediction. An early version of this work was presented at:

- 27th British Machine Vision Conference (BMVC2016) in York, UK.

Chapter 5 aims to give better understanding of the objects by tracing how the semantic categories evolve, and utilizes the CNN-RNN framework to fulfill the hierarchical image classification task. This framework can not only generate hierarchical labels for images, but also improve the traditional leaf-level classification performance by incorporating the relationship between hierarchical labels. In addition, we also investigate how we can utilize the framework to benefit the classification when a fraction of the training data is coarse-labeled. This work has been submitted to:

- Multimedia Tools and Applications (journal)

Chapter 6 focuses on a new emerging research area, i.e. image captioning, and investigates the effects of different Convnets. To obtain a richer visual representation, we propose aggregating their activations and achieve promising performance. This work has been published in the conference proceeding:

- 23rd International Conference on Multimedia Modeling (MMM2017) in Reykjavik, Iceland.

Chapter 7 concludes the thesis and reflects on our future work.

These are the publications which are related to the contents of this thesis:

- **Guo Y.**, Bai L., Lao S., Wu S., and Lew M.S., "A Comparison between Artificial Neural Network and Cascade-Correlation Neural Network in Concept Classification." 15th Pacific Rim Conference on Multimedia, 2014.
- **Guo Y.**, Lao S., Liu Y., Bai L., Liu S., and Lew M.S., "Convolutional Neural Networks Features: Principal Pyramidal Convolution." 16th Pacific Rim Conference on Multimedia, 2015.
- **Guo Y.**, and Lew M.S., "Bag of Surrogate Parts: one inherent feature of deep CNNs." 27th British Machine Vision Conference, 2016.
- Liu Y.*, **Guo Y.***, and Lew M.S., "What Convnets Make for Image Captioning?" 23rd International Conference on Multimedia Modeling, 2017 (* means equal contribution).
- **Guo Y.**, Liu Y., Oerlemans A, Lao S., Wu S., and Lew M.S., "Deep learning for visual understanding: A review." Neurocomputing, vol 187, 2016.
- **Guo Y.**, Liu Y., Lao S., Bakker E.M., Bai L., and Lew M.S., "Bag of Surrogate Parts for Visual Recognition." IEEE Transactions on Multimedia (submitted).
- **Guo Y.**, Liu Y., Bakker E.M., Guo Y., and Lew M.S., "CNN-RNN: A Large-scale Hierarchical Image Classification Framework." Multimedia Tools and Applications (submitted).
- Liu Y., **Guo Y.**, Wu S., and Lew M.S., "DeepIndex for Accurate and Efficient Image Retrieval." Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 2015.
- Liu Y., **Guo Y.**, and Lew M.S., "On the Exploration of Convolutional Fusion Networks for Visual Recognition." 23rd International Conference on

Multimedia Modeling, 2017 (Best Paper).

- Liu Y., **Guo Y.**, Bakker E.M., and Lew M.S., "Learning a Recurrent Residual Fusion Network for Multimodal Matching." International Conference on Computer Vision, 2017.

# Chapter 2

# A Comprehensive Review of Deep Learning Methods and Applications

Deep learning algorithms are a subset of the machine learning algorithms, which aim at discovering multiple levels of distributed representations. Recently, numerous deep learning algorithms have been proposed to solve traditional artificial intelligence problems. This chapter aims to review the state-of-the-art in deep learning algorithms in computer vision by highlighting the contributions and challenges from about 200 recent research papers. It first gives an overview of various deep learning approaches and their recent developments, and then briefly describes their applications in diverse vision tasks, such as image classification, object detection, image retrieval, semantic segmentation and human pose estimation. Finally, the chapter summarizes the future trends and challenges in designing and training deep neural networks.

## 2.1 Introduction

Deep learning is a subfield of machine learning which attempts to learn high-level abstractions in data by utilizing hierarchical architectures. It is an emerging approach and has been widely applied in traditional artificial intelligence domains, such as semantic parsing [11], natural language processing [12], computer vision [13, 14] and many more. There are mainly three important reasons for the booming of deep learning today: the dramatically increased chip processing abilities (e.g. GPU units), the significantly lowered cost of computing hardware, and the considerable advances in the machine learning algorithms [15].

Deep learning approaches have been extensively reviewed and discussed in recent years [15–19]. Among those Schmidhuber et al. [17] emphasized the important inspirations and technical contributions in a historical timeline format, while Bengio [18] examined the challenges of deep learning research and proposed a few forward-looking research directions. Deep networks have shown to be successful for computer vision tasks because they can extract appropriate features while jointly performing discrimination [15, 20]. In recent ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competitions [21], deep learning methods have been widely utilized by researchers and achieved top accuracy scores.

This chapter is intended to be useful to general neural computing, computer vision and multimedia researchers who are interested in state-of-the-art deep learning studies for computer vision. It provides an overview of various deep learning algorithms and their applications, especially those that can be applied in the computer vision domain.

The remainder of this chapter is organized as follows: In Section 2.2, we divide the deep learning algorithms into four categories: Convolutional Neural Networks, Restricted Boltzmann Machines, Autoencoder and Sparse Coding. Some well-known models in these categories as well as their developments are listed. We also describe the contributions and limitations for these models in this section. In Section 2.3, we describe the achievements of deep learning schemes in various computer vision applications, e.g. image classification, object detection, image retrieval, semantic segmentation and human pose estimation. The results on

these applications are shown and compared in the pipeline of their commonly used datasets. In Section 2.4, along with the success deep learning methods have achieved, we also face several challenges when designing and training the deep networks. In this section, we summarize some major challenges for deep learning, together with the inherent trends that might be developed in the future. In Section 2.5, we conclude this chapter.

## 2.2 Methods and recent developments

In recent years, deep learning has been extensively studied in the field of computer vision and as a consequence, a large number of related approaches have emerged. Generally, these methods can be divided into four categories according to the basic method they are derived from: Convolutional Neural Networks (CNNs), Restricted Boltzmann Machines (RBMs), Autoencoder and Sparse Coding.

The categorization of deep learning methods along with some representative works is shown in Figure 2.1.

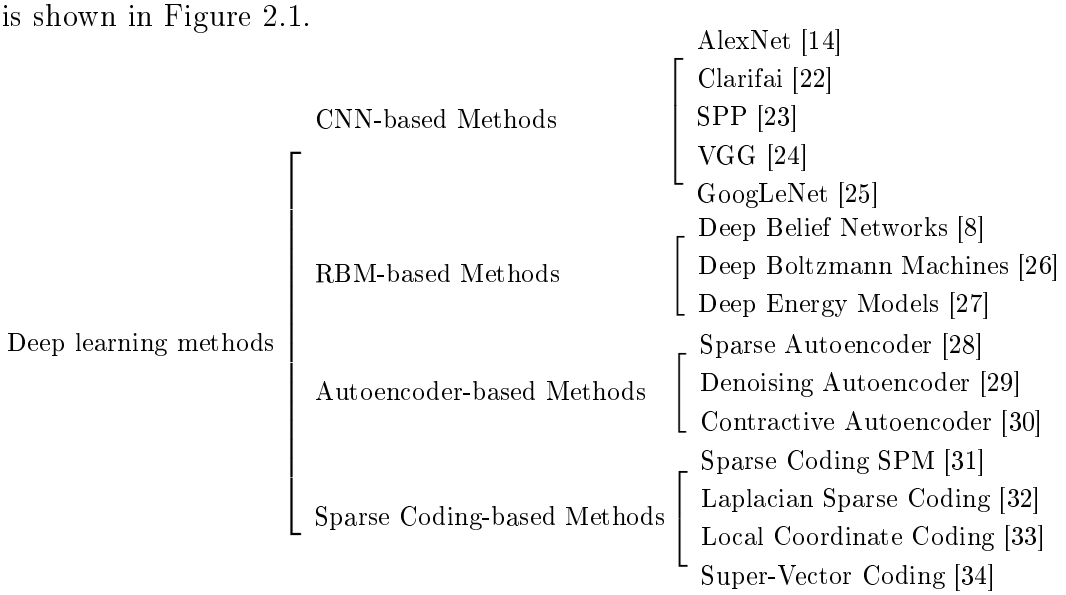



**Figure 2.1:** A categorization of the deep learning methods and their representative works.

In the next four parts, we will briefly review each of these deep learning methods and their most recent developments.

### 2.2.1 Convolutional Neural Networks (CNNs)

The Convolutional Neural Networks (CNN) is one of the most notable deep learning approaches where multiple layers are trained in an end-to-end manner [35]. It has been found highly effective and is also the most commonly used in diverse computer vision applications.

The pipeline of the general CNN architecture is shown in Figure 2.2.

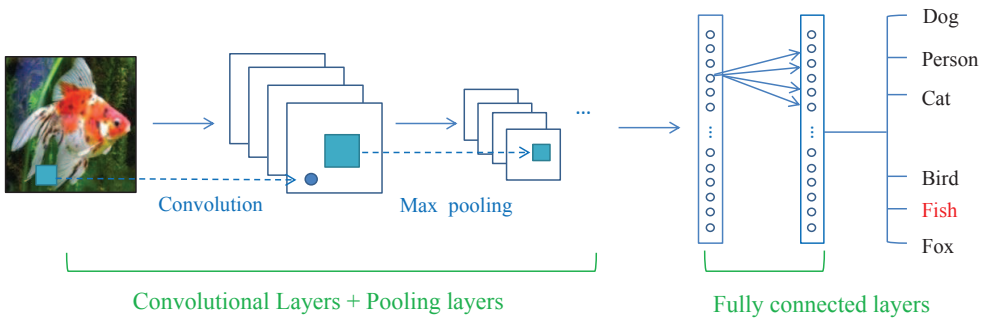



**Figure 2.2:** The pipeline of the general CNN architecture.

Generally, a CNN consists of three main neural layers, which are convolutional layers, pooling layers, and fully connected layers. Different kinds of layers play different roles. In Figure 2.2, a general CNN architecture for image classification [14] is shown layer-by-layer. There are two stages for training the network: a forward stage and a backward stage. First, the main goal of the forward stage is to represent the input image with the current parameters (weights and bias) in each layer. Then the prediction output is used to compute the loss cost with the ground truth labels. Second, based on the loss cost, the backward stage computes the gradients of each parameter with chain rules. All the parameters are updated based on the gradients, and are prepared for the next forward computation. After sufficient iterations of the forward and backward stages, the network learning can be stopped.

Next, we will first introduce the functions along with the recent developments of each layer, and then summarize the commonly used training strategies of the networks. Finally, we present several well-known CNN models, derived models, and conclude with the current tendency for using these models in real applications.

#### 2.2.1.1 Types of layers

Generally, a CNN is a hierarchical neural network whose convolutional layers alternate with pooling layers, followed by some fully connected layers (see Figure 2.2). In this section, we will present the function of the three layers and briefly review the recent advances that have appeared on those layers.

- **Convolutional layers**

In the convolutional layers, a CNN utilizes various kernels to convolve the whole image as well as the intermediate feature maps, generating various feature maps, as shown in Figure 2.3.

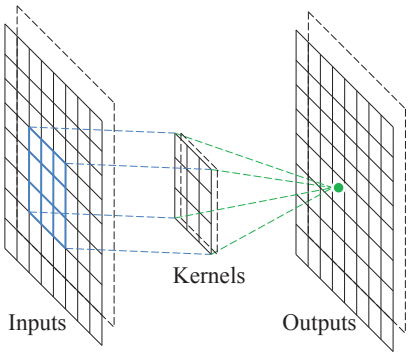



**Figure 2.3:** The operation of the convolutional layer.

There are three main advantages of the convolution operation [36]: 1) the weight sharing mechanism in the same feature map reduces the number of parameters; 2) local connectivity learns correlations among neighboring pixels; 3) invariance to the location of the object.

Due to the benefits introduced by the convolution operation, some well-known research papers use it as a replacement for the fully connected layers to accelerate the learning process [25, 37]. One interesting approach of handling the convolutional layers is the Network in Network (NIN) [38] method, where the main idea is to substitute the conventional convolutional layer with a small multilayer perceptron consisting of multiple fully connected layers with nonlinear activation functions, thereby replacing the linear filters with nonlinear neural networks. This method achieves good results in image classification.

- **Pooling layers**

Generally, a pooling layer follows a convolutional layer, and can be used to reduce the dimensions of feature maps and network parameters. Similar to convolutional layers, pooling layers are also translation invariant, because their computations take neighboring pixels into account. Average pooling and max pooling are the most commonly used strategies. Figure 2.4 gives an example for a max pooling process. For $8 \times 8$ feature maps, the output maps reduce to $4 \times 4$ dimensions, with a max pooling operator which has size $2 \times 2$ and stride 2.

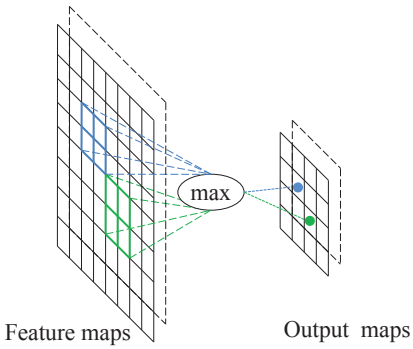



**Figure 2.4:** The operation of the max pooling layer.

For max pooling and average pooling, Boureau et al. [39] provided a detailed theoretical analysis of their performances. Scherer et al. [40] further conducted a comparison between the two pooling operations and found that max-pooling can lead to faster convergence, select superior invariant features and improve generalization. In recent years, various fast GPU implementations of CNN variants were presented, most of them utilize the max-pooling strategy [14, 41].

The pooling layers are the most extensively studied among the three layers. There are three well-known approaches related to the pooling layers, each having different purposes.

### √ Stochastic Pooling

A drawback of max pooling is that it is sensitive to overfit the training set, making it hard to generalize well to test samples [36]. Aiming to solve this problem, Zeiler et al. [42] proposed a stochastic pooling approach which replaces the conventional deterministic pooling operations with a stochastic procedure, by randomly picking the activation within each pooling region according to a multinomial distribution. It is equivalent to standard max pooling but with many copies of the input image, each having small local deformations. This stochastic nature is helpful to prevent the overfitting problem.

### √ Spatial Pyramid Pooling (SPP)

Normally, the CNN-based methods require a fixed-size input image. This restriction may reduce the recognition accuracy for images of arbitrary sizes. To eliminate this limitation, He et al. [23] utilized the general CNN architecture but replaced the last pooling layer with a spatial pyramid pooling layer. The spatial pyramid pooling can extract fixed-length representations from arbitrary images (or regions), generating a flexible solution for handling different scales, sizes, aspect ratios, and can be applied in any CNN structure to boost the performance of this structure.

### √ Def-Pooling

Handling deformation is a fundamental challenge in computer vision, especially for the object recognition task. Max pooling and average pooling are useful in handling deformation but they are not able to learn the deformation constraint and geometric model of object parts. To deal with deformation more efficiently, Ouyang et al. [43] introduced a new deformation constrained pooling layer, called def-pooling layer, to enrich the deep model by learning the deformation of visual patterns. It can substitute the traditional max-pooling layer at any information abstraction level.

Because of the different purposes and procedures the pooling strategies are designed for, various pooling strategies could be combined to boost the performance of a CNN.

- **Fully-connected layers**

Following the last pooling layer in the network as seen in Figure 2.2, there are several fully-connected layers converting the 2D feature maps into a 1D feature vector, for further feature representation, as seen in Figure 2.5.

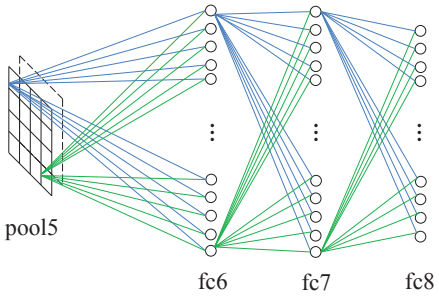



**Figure 2.5:** The operation of the fully-connected layer.

Fully-connected layers perform like a traditional neural network and contain about 90% of the parameters in a CNN. It enables us to feed forward the neural network into a vector with a pre-defined length. We could either feed forward the vector into certain number categories for image classification [14] or take it as a feature vector for follow-up processing [44].

Changing the structure of the fully-connected layer is uncommon, however an example came in the transferred learning approach [45], which preserved the parameters learned by ImageNet [14], but replaced the last fully-connected layer with two new fully-connected layers to adapt to the new visual recognition tasks.

The drawback of these layers is that they contain many parameters, which results in a large computational effort for training them. Therefore, a promising and commonly applied direction is to remove these layers or decrease the connections with a certain method. For example, GoogLeNet [25] designed a deep and wide network while keeping the computational budget constant, by switching from fully connected to sparsely connected architectures.

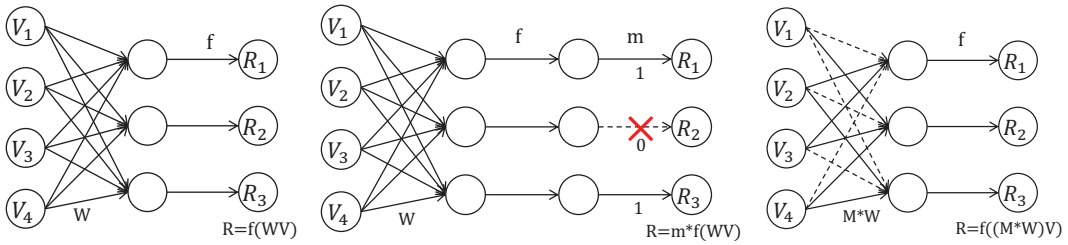
#### 2.2.1.2 Training Strategy

Compared to shallow learning, the advantage of deep learning is that it can build deep architectures to learn more abstract information. However, the large amount of parameters introduced may also lead to another problem: overfitting. Recently, numerous regularization methods have emerged in defense of overfitting, including the stochastic pooling mentioned above. In this section, we will introduce several other regularization techniques that may influence the training performance.

- **Dropout and DropConnect**

Dropout was proposed by Hinton et al. [46] and explained in-depth by Baldi et al. [47]. During each training case, the algorithm will randomly omit half of the feature detectors in order to prevent complex co-adaptations on the training data and enhance the generalization ability. This method was further improved in [48–53]. Specifically, Warde-Farley et al. [53] analyzed the efficacy of dropouts and suggested that dropout is an extremely effective ensemble learning method.

One well-known generalization derived from Dropout is called DropConnect [54], which randomly drops weights rather than the activations. Experiments showed that it can achieve competitive or even better results on a variety of standard benchmarks, although slightly slower. Figure 2.6 gives a comparison of No-Drop, Dropout and DropConnect networks [54].




(a) No-drop Network (b) Dropout Network (c) Dropconnect Network

**Figure 2.6:** A comparison of No-Drop, Dropout and DropConnect networks [54].

- **Data Augmentation**

When a CNN is applied to visual object recognition, data augmentation is often utilized to generate additional data without introducing extra labeling costs. The well-known AlexNet [14] employed two distinct forms of data augmentation: the first form of data augmentation consists of generating image translations and horizontal reflections, and the second form consists of altering the intensities of the RGB channels in training images. Howard et al. [55] took AlexNet as the base model and added additional transformations that improved the translation invariance and color invariance by extending image crops with extra pixels and adding additional color manipulations. This data augmentation method was widely utilized by some of the more recent studies [23, 25]. Dosovitskiy et al. [56] proposed an unsupervised feature learning approach based on data augmentation: it first randomly sampled a set of image patches and declares each of them as a surrogate class, then extended these classes by applying transformations corresponding to translation, scale, color and contrast. Finally, it trained a CNN to discriminate between these surrogate classes. The features learnt by the network showed good results on a variety of classification tasks. Aside from the classic methods such as scaling, rotating and cropping, Wu et al. [57] further adopted color casting, vignetting and lens distortion techniques, which produced more training examples with broad coverage.

- **Pre-training and fine-tuning**

Pre-training means to initialize the networks with pre-trained parameters rather than randomly set parameters. It is quite popular in models based on CNNs, due to the advantages that it can accelerate the learning process as well as improve the generalization ability. Erhan et al. [58] has conducted extensive simulations on the existing algorithms to find why pre-trained networks work better than networks trained in a traditional way. As AlexNet [14] achieved excellent performance and is released to the public, numerous approaches choose AlexNet trained on ImageNet2012 as their baseline deep model [23, 44, 45], and use fine-tuning of the parameters according to their specific tasks. Nevertheless, there are approaches [43, 59, 60] that deliver better performance by training on other models, e.g. Clarifai [22], GoogLeNet [25], and VGG [24].

Fine-tuning is a crucial stage for refining models to adapt to specific tasks and datasets. In general, fine-tuning requires class labels for the new training dataset, which are used for computing the loss functions. In this case, all layers of the new model will be initialized based on the pre-trained model, such as AlexNet [14], except for the last output layer that depends on the number of class labels of the new dataset and will therefore be randomly initialized. However, in some occasions, it is very difficult to obtain the class labels for any new dataset. To address this problem, a similarity learning objective function was proposed to be used as the loss functions without class labels [61], so the back-propagation can work normally and allow the model to be refined layer by layer. There are also many research results describing how to transfer the pre-trained model efficiently. A new way is defined to quantify the degree to which a particular layer is general or specific [62], namely how well features at that layer transfers from one task to another. They concluded that initializing a network with transferred features from almost any number of layers can give a boost to generalization performance after fine-tuning to a new dataset.

In addition to the regularization methods described above, there are also other common methods such as weight decay, weight tying and many more [17]. Weight decay works by adding an extra term to the cost function to penalize the parameters, preventing them from exactly modeling the training data and therefore helping to generalize to new examples [14]. Weight tying allows models to learn good representations of the input data by reducing the number of parameters in Convolutional Neural Networks [63].

One noteworthy thing is that these regularization techniques for training are not mutually exclusive and they can be combined to boost the performance.

#### 2.2.1.3 CNN architecture

With the recent developments of CNN schemes in the computer vision domain, some well-known CNN models have emerged. In this section, we first describe the commonly used CNN models, and then summarize their characteristics in their

applications. The configurations and the primary contributions of several typical CNN models are listed in Table 2.1.

**Table 2.1:** CNN models and their achievements in ILSVRC classification competitions.

| Method | Year | Place | Configuration | Contribution |
|---|---|---|---|---|
| AlexNet [14] | 2012 | $1^{st}$ | Five convolutional layers + three fully connected layers | an important CNN architecture which set the tone for many computer vision researches |
| Clarifai [22] | 2013 | $1^{st}$ | Five convolutional layers + three fully connected layers | insight into the function of intermediate feature layers |
| SPP [23] | 2014 | $3^{rd}$ | Five convolutional layers + three fully connected layers | proposed the 'spatial pyramid pooling' to remove the requirement of image resolution |
| VGG [24] | 2014 | $2^{nd}$ | Thirteen/Fifteen convolutional layers + three fully connected layers | a thorough evaluation of networks of increasing depth |
| GoogLeNet [25] | 2014 | $1^{st}$ | Twenty-one convolutional layers + one fully connected layer | increased the depth and width without raising the computational requirements |

AlexNet [14] is a significant CNN architecture, which consists of five convolutional layers and three fully connected layers. After inputting one fixed-size ($224 \times 224$) image, the network would repeatedly convolve and pool the activations, then forward the results into the fully-connected layers. The network was trained on ImageNet and integrated various regularization techniques, such as data augmentation, dropout, etc. AlexNet won the ILSVRC2012 competition [21], and set the tone for the surge of interest in deep convolutional neural networks. Nevertheless, there are two major drawbacks of this model: 1) it requires a fixed resolution of the image; 2) there is no clear understanding of why it performs so well.

In 2013, Zeiler et al. [22] introduced a novel visualization technique to give insight into the inner workings of the intermediate feature layers. These visualizations enabled them to find architectures that outperform AlexNet [14] on the ImageNet classification benchmark, and the resulting model, Clarifai, received top performance in the ILSVRC2013 competition.

As for the requirement of a fixed resolution, He et al. [23] proposed a new pooling strategy, i.e. spatial pyramid pooling, to eliminate the restriction of the image size. The resulting SPP-net could boost the accuracy of a variety of published CNN architectures despite their different designs.

In addition to the commonly used configuration of the CNN structure (five convolutional layers plus three fully connected layers), there are also approaches trying to explore deeper networks. In contrast to AlexNet, VGG [24] increased the depth of the network by adding more convolutional layers and taking advantage of very small convolutional filters in all layers. Similarly, Szegedy et al. [25] proposed a model, GoogLeNet, which also has quite a deep structure (22 layers) and has achieved leading performance in the ILSVRC2014 competition [21].

Despite top-tier classification performances have been achieved by various models, CNN-related models and applications are not limited to only image classification. Based on these models, new frameworks have been derived to address other challenging tasks, such as object detection, semantic segmentation, etc.

There are two well-known derived frameworks: RCNN (Regions with CNN features) [44] and FCN (fully convolutional network) [60], mainly designed for object detection and semantic segmentation respectively, as shown in Figure 2.7.

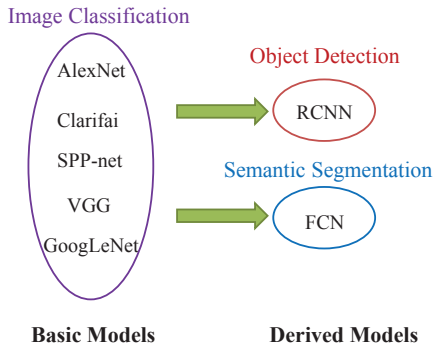



**Figure 2.7:** CNN basic models and derived models.

The core idea of RCNN is to generate multiple object proposals, extract features from each proposal using a CNN, and then classify each candidate window with a category-specific linear SVM. The "recognition using regions" paradigm received

encouraging performance in object detection and has gradually become the general pipeline for recent promising object detection algorithms [64–67]. However, the performance of RCNN relies too much on the precision of the object location, which may limit its robustness. Besides, the generation and processing of large number of proposals would also decrease its efficiency. Recent developments [64–66, 68, 69] are mainly focused on these two aspects.

RCNN takes the CNN models as feature extractor and does not make any change to the networks. In contrast, FCN proposes to recast the CNN models as fully convolutional nets, which removes the restriction of image resolution and could produce correspondingly-sized output efficiently. Although FCN is proposed mainly for semantic segmentation, the technique could also be utilized in other applications, e.g. image classification [70], edge detection [71] etc.

Aside from creating various models, the usage of these models also demonstrates several characteristics:

- **Large Networks**

One intuitive idea is to improve the performance of CNNs by increasing their sizes, which includes increasing the depth (the number of levels) and the width (the number of units at each level) [25]. Both aforementioned GoogLeNet [25] and VGG [24] utilized quite large networks, 22 layers and 19 layers respectively, demonstrating that increasing the size is beneficial for image recognition.

Jointly training multiple networks could lead to better performance than a single one. There are also many researchers [43, 72, 73] who designed large networks by combining different deep structures in cascade mode, where the output of the former networks is utilized by the latter ones, as shown in Figure 2.8.

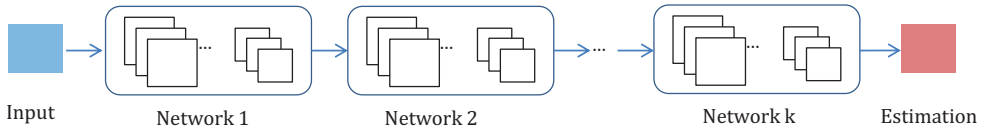



**Figure 2.8:** Combining deep structures in cascade mode.

The cascade architecture can be utilized to handle different tasks, and the function of the prior networks (i.e. the output) may vary with the tasks. For example, Wang et al. [73] connected two networks for object extraction, and the first network is used for object localization. Therefore, the output is the corresponding coordinates of the object. Sun et al. [72] proposed three-level carefully designed convolutional networks to detect facial keypoints. The first level provides highly robust initial estimations, while the following two levels fine-tune the initial prediction. Similarly, Ouyang et al. [43] adopted a multi-stage training scheme proposed by Zeng et al. [74], i.e. classifiers at the previous stages jointly work with the classifiers at the current stage to deal with misclassified samples.

- **Multiple Networks**

Another tendency in current applications is to combine the results of multiple networks, where each of the networks can execute the task independently, instead of designing a single architecture and jointly training the networks inside to execute the task, as seen in Figure 2.9.

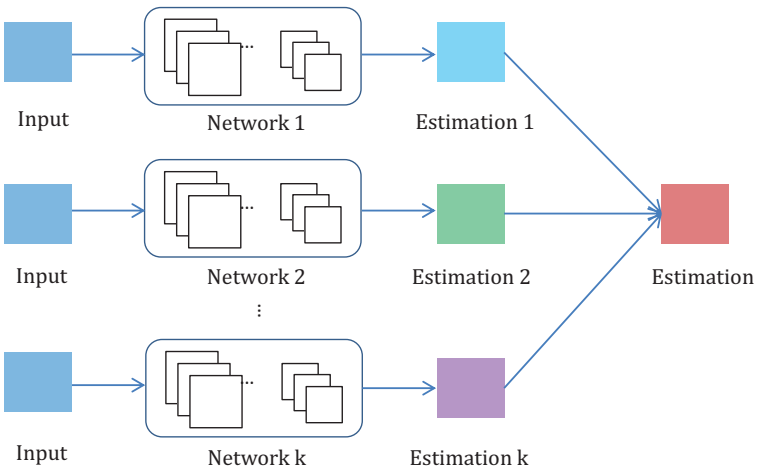



**Figure 2.9:** Combining the results of multiple networks.

Miclut et al. [75] gave some insight into how we should generate the final results when we have received a set of scores. Prior to the AlexNet [14], Ciresan et al. [13] proposed a method called Multi-Column DNN (MCDNN) which combines several DNN columns and averages their predictions. This model achieved

human-competitive results on tasks such as the recognition of handwritten digits or traffic signs. Recently, Ouyang et al. [43] also conducted an experiment to evaluate the performance of model combination strategies. It learnt 10 models with different settings and combined them in an averaging scheme. Results show that models generated in this way have high diversity and are complementary to each other in improving the detection results.

- **Diverse Networks**

Aside from altering the CNN structure, some researchers also attempt to introduce information from other sources, e.g. combining them with shallow structures, integrating contextual information, as illustrated in Figure 2.10.

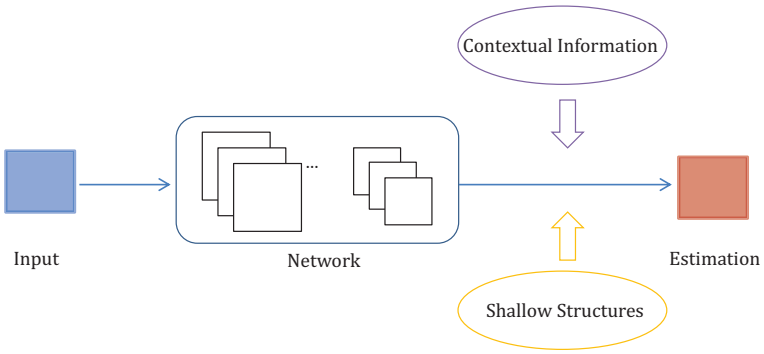



**Figure 2.10:** Combining a deep network with information from other sources.

Shallow methods can give additional insight into the problem. In the literature, examples can be found about combining shallow methods and deep learning frameworks [76], i.e. take a deep learning method to extract features and input these features to the shallow learning method, e.g. an SVM. One of the most representative and successful algorithms is the RCNN method [44], which feeds the highly distinctive CNN features into a SVM for the final object detection task. Besides, deep CNNs and Fisher Vectors (FV) are complementary [77] and can also be combined to significantly improve the accuracy of image classification.

Contextual information is sometimes available for an object detection task, and it is possible to integrate global context information with the information from the bounding box. In the ImageNet Large Scale Visual Recognition Challenge 2014

(ILSVRC 2014), the winning team NUS concatenated all the raw detection scores and combined them with the outputs from a traditional classification framework by context refinement [78]. Similarly, Ouyang et al. [43] also took the 1000-class image classification scores as the contextual features for object detection.

### 2.2.2 Restricted Boltzmann Machines (RBMs)

The Restricted Boltzmann Machine (RBM) is a generative stochastic neural network, and was proposed by Hinton et al. in 1986 [79]. It is a variant of the Boltzmann Machine, with the restriction that the visible units and hidden units must form a bipartite graph. This restriction allows for efficient training algorithms, in particular the gradient-based contrastive divergence algorithm [80].

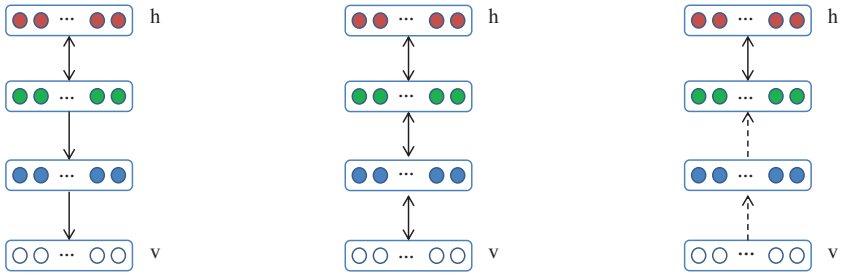
Since the model is a bipartite graph, the hidden units Hand the visible units $V_1$ are conditionally independent. Therefore,

$$P(H|V_1) = P(H_1|V_1)P(H_2|V_1)\cdots P(H_n|V_1) \tag{2.1}$$

Hinton [81] gave a detailed explanation and provided a practical way to train RBMs. Further work in [82] discussed the main difficulties of training RBMs, their underlying reasons and proposed a new algorithm, which consists of an adaptive learning rate and an enhanced gradient, to address those difficulties.

A well-known development of RBM can be found in [83]: the model approximates the binary units with noisy rectified linear units to preserve information about relative intensities as information travels through multiple layers of feature detectors. The refinement not only functions well in this model, but is also widely employed in various CNN-based approaches [14, 42].

Utilizing RBMs as learning modules, we can compose the following deep models: Deep Belief Networks (DBNs), Deep Boltzmann Machines (DBMs) and Deep Energy Models (DEMs). The comparison between the three models is shown in Figure 2.11.




(a) DBN1 (b) DBM1 (c) DEM1

**Figure 2.11:** The comparison of DBN, DBM and DEM [27].

DBNs have undirected connections at the top two layers which form an RBM and directed connections to the lower layers. DBMs have undirected connections between all layers of the network. DEMs have deterministic hidden units for the lower layers and stochastic hidden units at the top hidden layer [27].

A summary of these three deep models is provided in Table 2.2.

**Table 2.2:** An overview of representative RBM-based methods.

| Method | characteristics | advantages | drawbacks |
|---|---|---|---|
| DBN [8] | undirected connections at the top two layers and directed connections at the lower layers | 1.Properly initializes the network, which prevents poor local optima to some extent; 2.Training is unsupervised, which removes the necessity of labeled data for training | Due to the initialization process, it is computationally expensive to create a DBN model |
| DBM [26] | undirected connections between all layers of the network | Deals more robustly with ambiguous inputs by incorporating top-down feedback | The joint optimization is time-consuming |
| DEM [27] | deterministic hidden units for the lower layers and stochastic hidden units at the top hidden layer | Produces better generative models by allowing the lower layers to adapt to the training of higher layers | The learnt initial weight may not have good convergence |

In the next sections, we will explain these models and describe their applications to computer vision tasks respectively.

#### 2.2.2.1 Deep Belief Networks (DBNs)

The Deep Belief Network (DBN), proposed by Hinton [8], was a significant advance in deep learning. It is a probabilistic generative model which provides a joint probability distribution over observable data and labels. A DBN first takes advantage of an efficient layer-by-layer greedy learning strategy to initialize the deep network, and then fine-tunes all of the weights jointly with the desired outputs. The greedy learning procedure has two main advantages [84]: (1) it generates a proper initialization of the network, addressing the difficulty in parameter selection which may result in poor local optima to some extent; (2) the learning procedure is unsupervised and requires no class labels, so it removes the necessity of labeled data for training. However, creating a DBN model is a computationally expensive task that involves training several RBMs, and it is not clear how to approximate maximum-likelihood training to further optimize the model [19].

DBNs successfully focused researchers' attention on deep learning and as a consequence, many variants were created [85–88]. Nair et al. [88] developed a modified DBN where the top-layer model utilizes a third-order Boltzmann machine for object recognition. The model in [85] learned a two-layer model of natural images using sparse RBMs, in which the first layer learns local, oriented, edge filters, and the second layer captures a variety of contour features as well as corners and junctions. To improve the robustness against occlusion and random noise, Lee et al. [89] applied two strategies: one is to take advantage of sparse connections in the first layer of the DBN to regularize the model, and the other is to develop a probabilistic de-noising algorithm. When applied to computer vision tasks, a drawback of DBNs is that they do not consider the 2D structure of an input image. To address this problem, the Convolutional Deep Belief Network (CDBN) was introduced [86]. CDBN utilized the spatial information of neighboring pixels by introducing convolutional RBMs, generating a translation invariant generative model that scales well with high dimensional images. The algorithm was further extended in [90] and achieved excellent performance in face verification.

#### 2.2.2.2 Deep Boltzmann Machines (DBMs)

The Deep Boltzmann Machine (DBM), proposed by Salakhutdinov et al. [26], is another deep learning algorithm where the units are again arranged in layers. Compared to DBNs, whose top two layers form an undirected graphical model and whose lower layers form a directed generative model, the DBM has undirected connections across its structure.

Like the RBM, the DBM is also a subset of the Boltzmann family. The difference is that the DBM possesses multiple layers of hidden units, with units in odd-numbered layers being conditionally independent of even-numbered layers, and vice versa. Given the visible units, calculating the posterior distribution over the hidden units is no longer tractable, resulting from the interactions between the hidden units. When training the network, a DBM would jointly train all layers of a specific unsupervised model, and instead of maximizing the likelihood directly, the DBM uses a stochastic maximum likelihood (SML) [91] based algorithm to maximize the lower bound on the likelihood, i.e. performing only one or a few updates using a Markov chain Monte Carlo (MCMC) method between each parameter update. To avoid ending up in poor local minima which leave many hidden units effectively dead, a greedy layer-wise training strategy is also added into the layers when pre-training the DBM network, much in the same way as the DBN [19].

This joint learning has brought promising improvements, both in terms of likelihood and the classification performance of the deep feature learner. However, a crucial disadvantage of DBMs is the time complexity of approximate inference is considerably higher than DBNs, which makes the joint optimization of DBM parameters impractical for large datasets. To increase the efficiency of DBMs, some researchers introduced an approximate inference algorithm [92, 93], which utilizes a separate 'recognition' model to initialize the values of the latent variables in all layers, thus effectively accelerating the inference.

There are also many other approaches that aim to improve the effectiveness of DBMs. The improvements can either take place at the pre-training stage [94, 95] or at the training stage [96, 97]. For example, Montavon et al. [96] introduced

the centering trick to improve the stability of a DBM and made it to be more discriminative and generative. The multi-prediction training scheme [98] was utilized to jointly train the DBM which outperforms the previous methods in image classification proposed in [97].

#### 2.2.2.3 Deep Energy Models (DEMs)

The Deep Energy Model (DEM), introduced by Ngiam et al. [27], is a more recent approach to train deep architectures. Unlike DBNs and DBMs which share the property of having multiple stochastic hidden layers, the DEM just has a single layer of stochastic hidden units for efficient training and inference.

The model utilizes deep feed forward neural networks to model the energy landscape and is able to train all layers simultaneously. By evaluating the performance on natural images, it demonstrated the joint training of multiple layers yields qualitative and quantitative improvements over greedy layer-wise training. Ngiam et al. [27] used Hybrid Monte Carlo (HMC) to train the model. There are also other options including contrastive divergence, score matching, and others. A similar work can be found in [99].

Although RBMs are not as suitable as CNNs for vision applications, there are also some good examples adopting RBMs to vision tasks. The Shape Boltzmann Machine was proposed by Eslami et al. [100] to handle the task of modeling binary shape images, which learns high quality probability distributions over object shapes, for both realism of samples from the distribution and generalization to new examples of the same shape class. Kae et al. [101] combined the CRF and the RBM to model both local and global structure in face segmentation, which has consistently reduced the error in face labeling. A new deep architecture has been presented for phone recognition [102] that combines a Mean-Covariance RBM feature extraction module with a standard DBN. This approach attacks both the representational inefficiency issues of GMMs and an important limitation of previous work applying DBNs to phone recognition.

### 2.2.3 Autoencoder

The autoencoder is a special type of artificial neural network used for learning efficient encodings [103]. Instead of training the network to predict some target value Y given inputs X, an autoencoder is trained to reconstruct its own inputs X, therefore, the output vectors have the same dimensionality as the input vector. The general process of an autoencoder is shown in Figure 2.12.

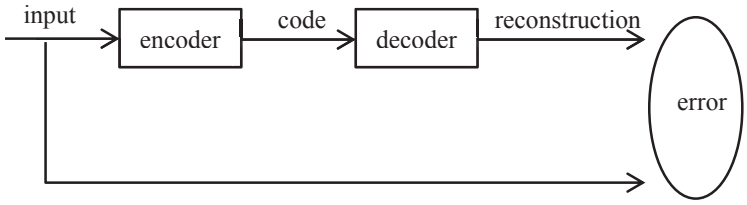



**Figure 2.12:** The pipeline of an autoencoder.

During the process, the autoencoder is optimized by minimizing the reconstruction error, and the corresponding code is the learned feature.

Generally, a single layer is not able to get the discriminative and representative features of raw data. Researchers now utilize the deep autoencoder, which forwards the code learnt from the previous autoencoder to the next, to accomplish their task.

The deep autoencoder was first proposed by Hinton et al. [104], and is still extensively studied in recent papers [105, 106]. A deep autoencoder is often trained with a variant of back-propagation, e.g. the conjugate gradient method. Though often reasonably effective, this model could become quite ineffective if errors are present in the first few layers. This may cause the network to learn to reconstruct the average of the training data. A proper approach to remove this problem is to pre-train the network with initial weights that approximate the final solution [104]. There are also variants of autoencoder proposed to make the representation as 'constant' as possible with respect to the changes in input.

In Table 2.3, we list some well-known variants of the autoencoder, and briefly summarize their characteristics and advantages. In the next sections, we de-

scribe three important variants: sparse autoencoder, denoising autoencoder and contractive autoencoder.

**Table 2.3:** Variants of the autoencoder.

| Method | characteristics | advantages |
|---|---|---|
| Sparse Autoencoder [28, 106] | Adds a sparsity penalty to force the representation to be sparse | 1. Make the categories to be more separable; 2. Make the complex data more meaningful; 3. In line with biological vision system |
| Denoising Autoencoder [29, 107] | Recovers the correct input from a corrupted version | More robust to noise |
| Contractive Autoencoder [30] | Adds an analytic contractive penalty to the reconstruction error function | Better captures the local directions of variation dictated by the data |
| Saturating Autoencoder [108] | Raises reconstruction error for inputs not near the data manifold | Limits the ability to reconstruct inputs which are not near the data manifold |
| Convolutional Autoencoder [109–111] | Shares weights among all locations in the input, preserving spatial locality | Utilizes the 2D image structure |
| Zero-bias Autoencoder [112] | Utilizes proper shrinkage function to train autoencoders without additional regularization | More powerful in learning representations on data with very high intrinsic dimensionality |

#### 2.2.3.1 Sparse Autoencoder

A sparse autoencoder aims to extract sparse features from raw data. The sparsity of the representation can either be achieved by penalizing the hidden unit biases [28, 85, 113] or by directly penalizing the output of hidden unit activations [114, 115].

Sparse representations have several potential advantages [28]: 1) using high-dimensional representations increases the likelihood that different categories will be easily separable, just as in the theory of SVMs; 2) sparse representations provide us with a simple interpretation of the complex input data in terms of a

number of 'parts'; 3) biological vision uses sparse representations in early visual areas [116].

A quite well-known variant of the sparse autoencoder is a nine-layer locally connected sparse autoencoder with pooling and local contrast normalization [117]. This model allows the system to train a face detector without having to label images as containing a face or not. The resulting feature detector is robust to translation, scaling and out-of-plane rotation.

#### 2.2.3.2 Denoising Autoencoder

In order to increase the robustness of the model, Vincent et al. [29, 107] proposed a model called denoising autoencoder (DAE), which can recover the correct input from a corrupted version, thus forcing the model to capture the structure of the input distribution. The process of a DAE is shown in Figure 2.13.

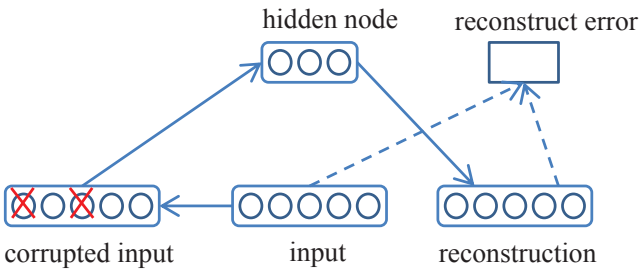



**Figure 2.13:** Denoising Autoencoder [29].

#### 2.2.3.3 Contractive Autoencoder

Contractive Autoencoder (CAE), proposed by Rifai et al. [30], followed after the DAE and shared a similar motivation of learning robust representations [19]. While a DAE makes the whole mapping robust by injecting noise in the training set, a CAE achieves robustness by adding an analytic contractive penalty to the reconstruction error function.

Although the differences between DAE and CAE are stated by Bengio et al. [19], Alain et al. [118] still suggested that DAE and a form of CAE are closely related

to each other: a DAE with small corruption noise can be valued as a type of CAE where the contractive penalty is on the whole reconstruction function rather than just on the encoder. Both DAE and CAE have been successfully used in the Unsupervised and Transfer Learning Challenge [119].

### 2.2.4 Sparse Coding

Sparse coding intends to learn an over-complete set of basic functions to describe the input data [120], and it has numerous advantages [31, 33, 121, 122]: (1) It can reconstruct the descriptor better by using multiple bases and capturing the correlations between similar descriptors which share bases; (2) the sparsity allows the representation to capture salient properties of images; (3) it is in line with the biological visual system, which argues that sparse features of signals are useful for learning; (4) image statistics study shows that image patches are sparse signals; (5) patterns with sparse features are more linearly separable.

#### 2.2.4.1 Solving the sparse coding equation

In this subsection, we will briefly describe how to solve the sparse coding problem. The general objective function of sparse coding is as below.

$$\min_{D} \frac{1}{T} \sum_{t=1}^{T} \min_{h^{(t)}} (\frac{1}{2} ||x^{(t)} - Dh^{(t)}||_2^2 + \lambda ||h^{(t)}||_1) \tag{2.2}$$

The first term of the function is the reconstruction error, while the second L1 regularization term is the sparsity penalty. The L1 norm regularization has been verified to lead to sparse representations [123]. Eq 2.2 can be solved with a regression method called LASSO (Least Absolute Shrinkage and Selection Operator). It cannot get the analytic solution of the sparse representation. Therefore, solving of the problem normally results in an intractable computation.

To optimize the sparse coding model, there is an alternating procedure between updating the weights and inferring the feature activations of the input given the current setting of the weights.

1. Weight update

One commonly used algorithm for updating the weights is called projected gradient algorithm [124], which renormalizes each column of the weight matrix right after each update of the traditional Gradient descent algorithm [125]. The normalization is necessary for the sparsity penalty to have any effect. However, gradient descent using iterative projections often shows slow convergence. In 2007, Lee et al. [126] derived a Lagrange dual method, which is much more efficient than gradient-based methods. Given a dictionary, the paper further proposed a feature-sign search algorithm to learn the sparse representation. The combination of these two algorithms enabled the performance to be significantly better than the previous ones. However, it cannot efficiently handle very large training sets, or dynamic training data that is changing over time. Thus it inherently accesses the whole training set at each iteration. To address this issue, an online approach [127, 128] was proposed for learning dictionaries that process one element (or a small subset) of the training set at a time. The algorithm then updates the dictionary using block-coordinate descent [129] with warm restarts, which does not require any learning rate tuning.

Gregor et al. [130] tried to accelerate the dictionary learning in another way: it imports the idea of Coordinate Descent algorithm (CoD) which only updates the "most promising" hidden units and therefore leads to dramatic reduction in the number of iterations to reach a given code prediction error.

2. Activation inference

Given a set of the weights, we need to infer the feature activations. A popular algorithm for sparse coding inference is the Iterative Shrinkage-Thresholding Algorithm (ISTA) [131], which takes a gradient step to optimize the reconstruction term, followed by a sparsity term which has a closed form shrinkage operation. Although simple and effective, the algorithm suffers from a severe problem that it converges quite slowly. The problem is partly solved by the Fast Iterative shrinkage-Thresholding Algorithm (FISTA) approach [132], which preserves the computational simplicity of ISTA, but converges more quickly due to the introduction of a 'momentum' term in the dynamics (the convergence complexity

changed from to ). Both the ISTA and FISTA inference involve some sort of iterative optimization (i.e. LASSO), which is of high computational complexity. In contrast, Kavukcuoglu et al. [133] utilized a feed-forward network to approximate the sparse codes, which dramatically accelerated the inference process. Furthermore, the LASSO optimization stage was replaced by marginal regression in [134], effectively scaling up the sparse coding framework to large dictionaries.

#### 2.2.4.2 Developments

As we have briefly stated how to generate the sparse representation given the objective function, in this subsection, we will introduce some well-known algorithms related to sparse coding, in particular those that are used in computer vision tasks. The well-known sparse coding algorithms and relations, along with their contributions and drawbacks are shown in Figure 2.14.

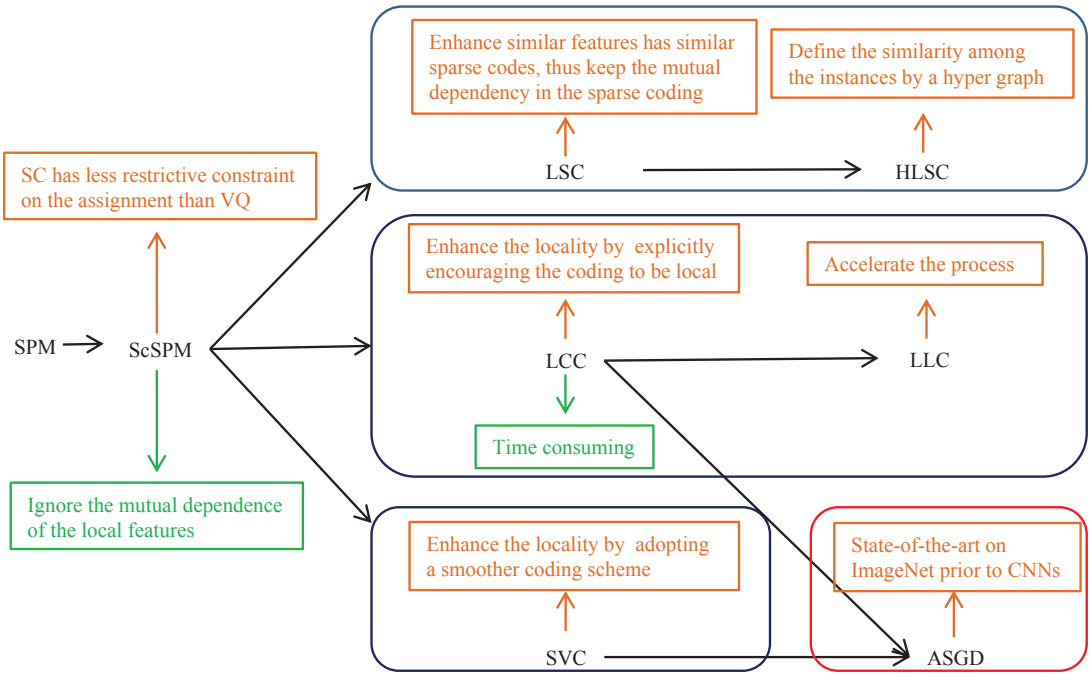
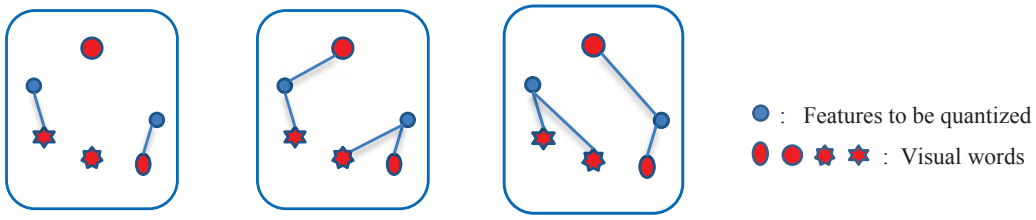



**Figure 2.14:** The well-known sparse coding algorithms, relations, contributions and drawbacks

One representative algorithm for sparse coding is Sparse coding SPM (ScSPM) [31], which is an extension of the Spatial Pyramid Matching (SPM) method [135]. Unlike the SPM, which uses vector quantization (VQ) for the image representation, ScSPM utilizes sparse coding (SC) followed by multi-scale spatial max pooling. The codebook of SC is an over-complete basis and each feature can activate a small number of them. Compared to VQ, SC receives a much lower reconstruction error due to the less restrictive constraint on the assignment. Coates et al. [136] further investigated the reasons for the success of SC over VQ in detail. A drawback of ScSPM is that it deals with local features separately, thus ignores the mutual dependence among them, which makes it too sensitive to feature variance, i.e. the sparse codes may vary a lot, even for similar features.

To address this problem, Gao et al. [32] proposed a Laplacian Sparse Coding (LSC) approach, in which similar features are not only assigned to optimally-selected cluster centers, but that also guarantees the selected cluster centers to be similar. The difference between K-means, Sparse Coding and Laplacian Sparse Coding is shown in Figure 2.15.




(a) K-means (b) Sparse Coding (c) Laplacian SC

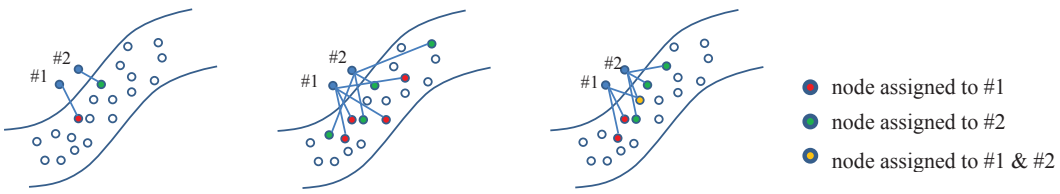
**Figure 2.15:** The difference between K-means, Sparse Coding and Laplacian Sparse Coding [32].

By adding the locality preserving constraint to the objective of sparse coding, the LSC can keep the mutual dependency in the sparse coding procedure. Gao et al. [137] further raised a Hyper-graph Laplacian Sparse Coding (HLSC) method, which extends the LSC to the case where the similarity among the instances is defined by a hyper graph. Both LSC and HLSC enhance the robustness of sparse coding.

Another way to address the sensitivity problem is the hierarchical sparse coding method proposed by Yu et al. [138]. It introduced a two-layer sparse coding model: the first layer encodes individual patches, and the second layer jointly encodes the set of patches that belong to the same group. Therefore, the model leverages the spatial neighborhood structure by modeling the higher-order dependency of patches in the same local region of an image. Besides that, it is a fully automatic method to learn features from the pixel level, rather than for example the hand-designed SIFT feature. The hierarchical sparse coding is utilized in another research [139] to learn features for images in an unsupervised fashion. The model is further improved by Zeiler et al. [140].

In addition to the sensitivity, another method exists for improving the ScSPM algorithm, by considering the locality. Yu et al. [33] observed that the ScSPM results tend to be local, i.e. nonzero coefficients are often assigned to bases nearby. As a result of these observations, they suggested a modification to ScSPM, called Local Coordinate Coding (LCC), which explicitly encourages the coding to be local. They also theoretically showed that locality is more important than sparsity. Experiments have shown that locality can enhance sparsity and that sparse coding is helpful for learning only when the codes are local, so it is preferred to let similar data have similar non-zero dimensions in their codes. Although LCC has a computational advantage over classic sparse coding, it still needs to solve the L1-norm optimization problem, which is time-consuming. To accelerate the learning process, a practical coding method called Locality-Constrained Linear Coding (LLC) was introduced [122], which can be seen as a fast implementation of LCC that replaces the L1-norm regularization with L2-norm regularization.

A comparison between VQ, ScSPM and LLC [122] are shown in Figure 2.16.




(a) VQ (b) ScSPM (c) LLC

**Figure 2.16:** A comparison between VQ, ScSPM, LLC [122].

Besides LLC, there is another model, called super-vector coding (SVC) [34], which can also guarantee local sparse coding. Given x, SVC will activate those coordinates associated to the neighborhood of x to achieve the sparse representation. SVC is a simple extension of VQ by expanding VQ in local tangent directions, and is thus a smoother coding scheme.

A remarkable result is shown in [141], in which the proposed averaging stochastic gradient descent (ASGD) scheme combined LCC and SVC algorithm to scale the image classification to large-scale dataset, and produced state-of-the-art results on ImageNet object recognition tasks prior to the rise of CNN architectures.

Another well-known smooth coding method is presented in [134], called Smooth Sparse Coding (SSC). The method incorporates the neighborhood similarity and temporal information into sparse coding, leading to codes that represent a neighborhood rather than an individual sample and that have lower mean square reconstruction error.

More recently, He et al. [142] proposed a new unsupervised feature learning framework, called Deep Sparse Coding (DeepSC), which extends sparse coding to a multi-layer architecture and has the best performance among the sparse coding schemes described above.

### 2.2.5 Discussion

In order to compare and understand the above four categories of deep learning, we summarize their advantages and disadvantages with respect to diverse properties, as listed in Table 2.4. There are nine properties in total. In details, 'Generalization' refers to whether the approach has been shown to be effective in diverse media (e.g. text, images, audio) and applications, including speech recognition, visual recognition and so on. 'Unsupervised learning' refers to the ability to learn a deep model without supervisory annotation. 'Feature learning' is the ability to automatically learn features based on a data set. 'Real-time training' and 'Real-time prediction' refer to the efficiency of the learning and inferring processes, respectively. 'Biological understanding' and 'Theoretical justification' represent

whether the approach has significant biological underpinnings or theoretical foundations, respectively. 'Invariance' refers to whether the approach has been shown to be robust to transformations such as rotation, scale and translation. 'Small training set' refers to the ability to learn a deep model using a small number of examples. It is important to note that the table only represents the general current findings and not future possibilities nor specialized niche cases.

**Table 2.4:** Comparisons among four categories of deep learning (Note: 'Yes' indicates that the category does well in the property; otherwise, they will be marked by 'No'. The 'Yes*' refers to a preliminary or weak ability)

| Properties | CNNs | RBMs | AutoEncoder | Sparse Coding |
|---|---|---|---|---|
| Generalization | Yes | Yes | Yes | Yes |
| Unsupervised learning | No | Yes | Yes | Yes |
| Feature learning | Yes | Yes* | Yes* | No |
| Real-time training | No | No | Yes | Yes |
| Real-time prediction | Yes | Yes | Yes | Yes |
| Biological understanding | No | No | No | Yes |
| Theoretical justification | Yes* | Yes | Yes | Yes |
| Invariance | Yes* | No | No | Yes |
| Small training set | Yes* | Yes* | Yes | Yes |

## 2.3 Applications and Results

Deep learning has been widely adopted in various directions of computer vision, such as image classification, object detection, image retrieval, semantic segmentation, and human pose estimation, which are key tasks for image understanding. In this part, we will briefly summarize the developments of deep learning (all of the results are referred from the original papers), especially the CNN based algorithms, in these five areas.

### 2.3.1 Image Classification

The image classification task consists of labeling input images with a probability of the presence of a particular visual object class [143], as is shown in Figure 2.17.

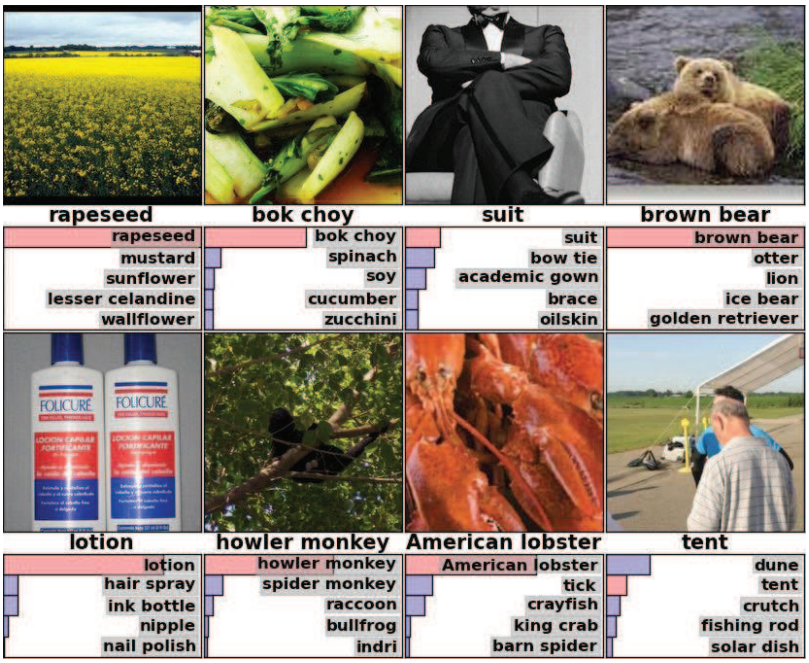



**Figure 2.17:** Image classification examples from AlexNet [14]. Each image has one ground truth label, followed by the top 5 guesses with probabilities.

Prior to deep learning, perhaps the most commonly used methods in image classification were methods based on bags of visual words (BoW) [144], which first describes the image as a histogram of quantized visual words, and then feeds the histogram into a classifier (typically an SVM [145]). This pipeline was based on the orderless statistics, to incorporate spatial geometry into the BoW descriptors. Lazebnik et al. [135] integrated a spatial pyramid approach into the pipeline, which counts the number of visual words inside a set of image sub-regions instead of the whole region. Thereafter, this pipeline was further improved by importing sparse coding optimization to the building of codebooks [141], which receives the

best performance on the ImageNet 1000-class classification in 2010. Sparse coding is one of the basic algorithms in deep learning, and it is more discriminative than the original hand-designed ones, e.g. HOG [141] and LBP [146].

The approaches based on BoW just concern the zero order statistics (i.e. counts of visual words), discarding a lot of valuable information of the image [143]. The method introduced by Perronnin et al. [147] overcame this issue and extracted higher order statistics by employing the Fisher Kernel [148], achieving the state-of-the-art image classification result in 2011.

Krizhevsky et al. [14] represented a turning point for large-scale object recognition when a large CNN was trained on the ImageNet database [149], thus proving that CNN could, in addition to handwritten digit recognition [35], perform well on natural image classification. The proposed AlexNet won the ILSVRC 2012, with a top-5 error rate of 15.3%, which sparked significant additional activity in CNN research. In Figure 2.18, we present the state-of-the-art results on the ImageNet test dataset since 2012, along with the pipeline of ILSVRC.

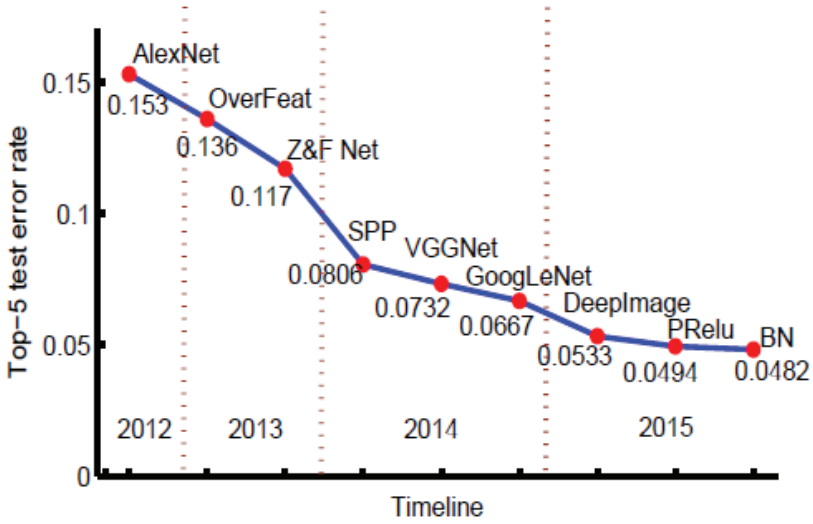



**Figure 2.18:** ImageNet classification results on test dataset.

OverFeat [150] proposed a multiscale and sliding window approach, which could find the optimal scale of the image and fulfill different tasks simultaneously, i.e.

classification, localization and detection. Specifically, the algorithm decreased the top-5 test error to 13.6%. Zeiler et al. [22] introduced a novel visualization technique to give insight into the function of intermediate feature layers and further adjusted a new model, which outperformed AlexNet, reaching 11.7% top-5 error rate, and had top performance at ILSVRC 2013.

ILSVRC 2014 witnessed the steep growth of deep learning, as most participants utilized CNNs as the basis for their models. Again significant progress had been made in image classification, as the error was almost halved since ILSVRC2013. The SPP-net [23] model eliminated the restriction of the fixed input image size and could boost the accuracy of a variety of published CNN architectures despite their different designs. Multiple SPP-nets further reduced the top-5 error rate to 8.06% and ranked third in the image classification challenge of ILSVRC 2014. Along with the improvements of the classic CNN model, another characteristic shared by the top-performing models is that the architectures became deeper, as shown by GoogLeNet [25] (rank 1 in ILSVRC 2014) and VGG [24] (rank 2 in ILSVRC 2014), which achieved 6.67% and 7.32% respectively.

Despite the potential capacity possessed by larger models, they also suffered from overfitting and underfitting problems when there is little training data or little training time. To avoid this shortcoming, Wu et al. [57] developed new strategies, i.e. DeepImage, for data augmentation and usage of multi-scale images. They also built a large supercomputer for deep neural networks and developed a highly optimized parallel algorithm, and the classification result achieved a relative 20% improvement over the previous one with a top-5 error rate of 5.33%. More Recently, He et al. [9] proposed the Parametric Rectified Linear Unit to generate the traditional rectified activation units and derived a robust initialization method. This scheme led to 4.94% top-5 test error and surpassed human-level performance (5.1%) for the first time. Similar results were achieved by Ioffe et al. [151], whose method reached a 4.8% test error by utilizing an ensemble of batch-normalized networks.

### 2.3.2 Object Detection

Object detection is different from but closely related to an image classification task. For image classification, the whole image is utilized as the input and the class label of objects within the image are estimated. For object detection, besides outputting the information of the presence of a given class, we also need to estimate the position of the instance (or instances), as shown in Figure 2.19. A detection window is regarded as correct if the outputted bounding box has sufficiently large overlap with the ground truth object (usually more than 50%).


**Figure 2.19:** Object detection examples from RCNN [44]. The red box extracts the salient objects contains, the green box contains the prediction score.

The challenging PASCAL VOC datasets are the most widely employed for the evaluation of object detection. There are twenty classes in this database. During the test phase, an algorithm should predict the bounding boxes of the objects belong to each class in a test image. In this section, we will describe the recent developments of deep learning schemes for object detection, according to their achievements in VOC 2007 and VOC 2012. The related advances are shown in Table 2.5.

Before the surge of deep learning, the Deformable Part Model (DPM) [152] was the most effective method for object detection. It takes advantage of deformable part models and detects objects across all scales and locations on the image in an exhaustive manner. After integrating with some post-processing techniques, i.e. bounding box prediction and context rescoring, the model achieved 29.09% average precision for VOC 2007 test set.

**Table 2.5:** Object detection results of the VOC 2007 and VOC 2012 challenges

| Methods | Training data | VOC2007 (mAP) | | VOC2012(mAP) | |
|---|---|---|---|---|---|
| | | A/C-Net | VGG-Net | Alex-Net | VGG-Net |
| DPM [152] | 07 | 29.09% | - | - | - |
| DetectorNet [153] | 12 | 30.41% | - | - | - |
| DeepMultiBox [154] | 12 | 29.22% | - | - | - |
| RCNN [44] | 07 | 54.2% | 62.2% | - | - |
| RCNN [44] + BB | 07 | 58.5% | 66% | - | - |
| RCNN [44] | 12 | - | - | 49.6% | 59.2% |
| RCNN [44] + BB | 12 | - | - | 53.3% | 62.4% |
| SPP-Net [23] | 07 | 55.2% | 60.4% | - | - |
| SPP-Net [23] | 07+12 | - | 64.6% | - | - |
| SPP-Net [23] + BB | 07 | 59.2% | 63.1% | - | - |
| FRCN [64] | 07 | - | 66.9% | - | 65.7% |
| FRCN [64] | 07++12 | - | 70.0% | - | 68.4% |
| RPN [65] | 07 | 59.9% | 69.9% | - | - |
| RPN [65] | 12 | - | - | - | 67% |
| RPN [65] | 07+12 | - | 73.2% | - | - |
| RPN [65] | 07++12 | - | - | - | 70.4% |
| MR_CNN [67] | 07 | - | 74.9% | - | 69.1% |
| MR_CNN [67] | 12 | - | - | - | 70.7% |
| FGS [69] | 07 | - | 66.5% | - | - |
| FGS [69] + BB | 07 | - | 68.5% | - | 66.4% |
| NoC [155] | 07+12 | 62.9% | 71.8% | - | 67.6% |
| NoC [155] + BB | 07+12 | - | 73.3% | - | 68.8% |

Note: Training data: "07": VOC07 trainval; "12": VOC2 trainval; "07+12": VOC07 trainval union with VOC12 trainval; "07++12": VOC07 trainval and test union with VOC12 trainval; BB: bounding box regression; A/C-Net: approaches based on AlexNet[6] or Clarifai [52]; VGG-Net: approaches based on VGG-Net[31]

As deep learning methods (especially the CNN-based methods) had achieved top tier performance on image classification tasks, researchers started to transfer it to the object detection problem. An early deep learning approach for object detection was introduced by Szegedy et al. [153]. The paper proposed an algorithm,

called DetectorNet, which replaced the last layer of AlexNet [14] with a regression layer. The algorithm captured object location well and achieved competitive results on the VOC2007 test set with the most advanced algorithms at that time. To handle multiple instances of the same object in the image, DeepMultiBox [154] also showed a saliency-inspired neural network model.

A general pattern for current successful object detection systems is to generate a large pool of candidate boxes and classify those using CNN features. The most representative approach is the RCNN scheme proposed by Girshick et al. [44]. It utilizes selective search [156] to generate object proposals, and extracts the CNN features for each proposal. The features are then fed into an SVM classifier to decide whether the related candidate windows contain the object or not. RCNNs improved the benchmark by a large margin, and became the base model for many other promising algorithms [64–66, 68, 69].

The algorithms derived from RCNNs are mainly divided into two categories: the first category aims to accelerate the training and testing process. Although an RCNN has excellent object detection accuracy, it is computationally intensive because it first warps and then processes each object proposal independently. Consequently, some well-known algorithms which aim to improve its efficiency appeared, such as SPP-net [23], FRCN [64], RPN [65], YOLO [157], etc. These algorithms detect objects faster, while achieving comparable mAP with state-of-the-art benchmarks.

The second category is mainly intended to improve the accuracy of RCNNs. The performance of the 'recognition using regions' paradigm is highly dependent on the quality of object hypotheses. Currently, there are many object proposal algorithms, such as objectness [158], selective search [156], category-independent object proposals [159], BING [160], and edge boxes [161]. These schemes are exhaustively evaluated in [162]. Although those schemes are good at finding rough object positions, they normally could not accurately localize the whole object via a tight bounding box, which forms the largest source of detection error [163, 164]. Therefore, many approaches have emerged that try to correct the poor localizations.

One important direction of these methods is to combine them with semantic segmentation techniques [66, 68, 165]. For example, the SDS scheme proposed by Hariharan et al. [68] utilizes segmentation to mask-out the background inside the detection, resulting in improved performance for object detection (from 49.6% to 50.7%, both without bounding box regression). On the other hand, the UDS method [165] unified the object detection and semantic segmentation process in one framework, by enforcing their consistency and integrating context information, the model demonstrated encouraging performance on both tasks. Similar works come with segDeepM proposed by Zhu et al. [66] and MR_CNN in [67], which also incorporate the segmentation along with additional evidence to boost the accuracy of object detection.

There are also approaches which attempt to precisely locate the object in other ways. For instance, FGS [69] addresses the localization problem via two methods: 1) develop a fine-grained search algorithm to iteratively optimize the location; 2) train a CNN classifier with a structured SVM objective to balance between classification and localization. The combination of these methods demonstrates promising performance on two challenging datasets.

Aside from the efforts in object localization, the NoC framework in [155] tries to evolve efforts in the object classification step. In place of the commonly used multi-layer perceptron (MLP), it explored different NoC structures to implement the object classifiers.

It is much cheaper and easier to collect a large amount of image-level labels than it is to collect detection data and label it with precise bounding boxes. Therefore, a major challenge in scaling the object detection is the difficulty of obtaining labeled images for large numbers of categories [166, 167]. Hoffman et al. [166] proposed a Deep Detection Adaption (DDA) algorithm to learn the difference between image classification and object detection, transferring classifiers for categories into detectors, without bounding box annotated data. The method has the potential to enable the detection for thousands of categories which lack bounding box annotations.

Two other promising, scalable approaches are ConceptLeaner [168] and BabyLearning [169].Both of them can learn accurate concept detectors but without the massive annotation of visual concepts. As collecting weakly labeled images is cheap, ConceptLeaner [168] develops a max-margin hard instance learning algorithm to automatically discover visual concepts from noisy labeled image collections. As a result, it has the potential to learn concepts directly from the web. On the other hand, the BabyLearning [169] approach simulates a baby's interaction with the physical world, and can achieve comparable results with state-of-the-art full-training based approaches with only few samples for each object category, along with large amounts of online unlabeled videos.

From Table 2.5, we can also observe several factors that could improve the performance, in addition to the algorithm itself: 1) larger training set; 2) deeper base model; 3) Bounding Box regression.

### 2.3.3 Image Retrieval

Image retrieval aims to find images containing a similar object or scene as in the query image, as illustrated in Figure 2.20.

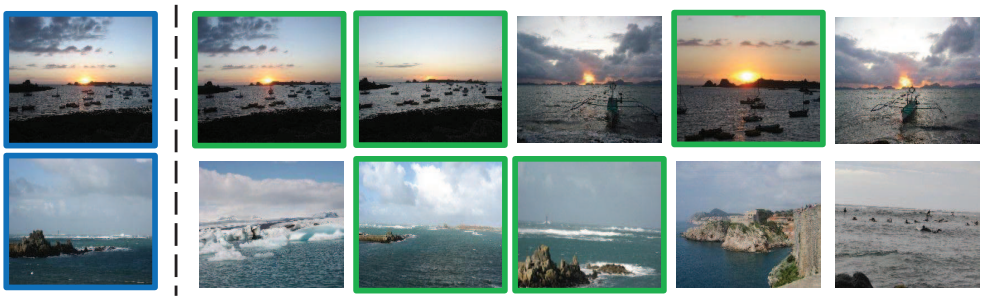


**Figure 2.20:** Image retrieval examples using CNN features. The left images are querying ones, and the images with green frames in the right represent the positive retrieval candidates.

The success of AlexNet [14] suggests that the features emerging in the upper layers of the CNN learned to classify images can serve as good descriptors for

image classification. Motivated by this, many recent studies use CNN models for image retrieval tasks [61, 170–173]. These studies achieved competitive results compared with the traditional methods, such as VLAD and Fisher Vector. In the following paragraphs, we will introduce the main ideas of these CNN based methods.

Inspired by Spatial Pyramid Matching, Gong et al. [170] proposed a kind of 'reverse SPM' idea that extracts patches at multiple scales, starting with the whole image, and then pool each scale without regard to spatial information. Then it aggregates local patch responses at the finer scales via VLAD encoding. The orderless nature of VLAD helps to build a more invariant representation. Finally, the original global deep activations are concatenated with the VLAD features for the finer scales to form the new image representation.

Razavian et al. [171] used features extracted from the OverFeat network as a generic image representation to tackle the diverse range of vision tasks, including recognition and retrieval. First, it augments the training set by adding cropped and rotated samples. Then for each image, it extracts multiple sub-patches of different sizes at different locations. Each sub-patch is computed for its CNN presentation. The distance between the reference and the query image is set to the average distance of each query sub-patch to the reference image.

Given the recent successes that deep learning techniques have achieved, the research presented in [61] attempts to evaluate if deep learning can bridge the semantic gap in content-based image retrieval (CBIR). Their encouraging results reveal that deep CNN models pre-trained on large datasets can be directly used for feature extraction in new CBIR tasks. When being applied for feature representation in a new domain, it was found that similarity learning can further boost the retrieval performance. Further, by retraining the deep models with a classification or similarity learning objective on the new domain, the accuracy can be improved significantly.

A different approach shown in [172] is to first extract object-like image patches with a general object detector. Then, one CNN feature is extracted in each object patch with the pre-trained AlexNet model. With many results from their

experiments, it is concluded that their method can achieve a significant accuracy improvement with the same space consumption, and with the same time cost it still obtains a higher accuracy.

Finally, without sliding windows or multiple-scale patches, Babenko et al. [173] focus on holistic descriptors where the whole image is mapped to a single vector with a CNN model. It found that the best performance is observed not at the very top of the network, but rather at the layer that is two levels below the outputs. An important result is that PCA affects the performance of the CNN much less than the performance of VLADs or Fisher Vectors. Therefore PCA compression works better for CNN features. In Table 2.6, we show the retrieval results in several public datasets.

**Table 2.6:** Image retrieval results on several datasets

| Methods | Holidays | Paris6K | Oxford5K | UKB |
|---|---|---|---|---|
| Babenko et al. [173] | 74.7 | - | 55.7 | 3.43 |
| SUN et al. [172] | 79.0 | - | - | 3.61 |
| Gong et al. [170] | 80.2 | - | - | - |
| Razavian et al. [171] | 84.3 | 79.50 | 68.0 | -(91.1) |
| Wan et al. [61] | - | 94.7 | 78.3 | - |

There is one more interesting problem in CNN features: which layer has the highest impact on the final performance? Some methods extract features in the second fully connected layer [170, 172]. In contrast to them, other methods use the first fully connected layer in their CNN model for image representation [171, 173]. Moreover, these choices may change for different datasets [61]. Thus, we think investigating the characteristics of each layer is still an open problem.

### 2.3.4 Semantic Segmentation

In recent years, a large number of studies focus on the semantic segmentation task, and yield promising progress. The main reason of their success comes from CNN models, which are capable of tackling the pixel-level predictions with the

pre-trained networks on large-scale datasets. Different from image-level classification and object-level detection, semantic segmentation requires output masks that have a 2D spatial distribution. As for semantic segmentation, recent and advanced CNN based methods can be summarized as follows:

(1) Detection-based segmentation. The approach segments images based on the candidate windows outputted from object detection [44, 174–176]. RCNN [44] and SDS [68] first generated region proposals for object detection, and then utilized traditional approaches to segment the region and to assign the pixels with the class label from detection. Based on SDS [68], Hariharan et al. [176] proposed the hyper-column at each pixel as the vector of activations, and gained large improvement. One disadvantage of detection-based segmentation is the largely additional expense for object detection. Without extracting regions from raw images, Dai et al. [175] designed a convolutional feature masking (CFM) method to extract proposals directly from the feature maps, which is efficient as the convolutional feature maps only need to be computed once. Even though, the errors caused by proposals and object detection tend to be propagated to the segmentation stage.

(2) FCN-CRFs based segmentation. In the second one, fully convolutional networks(FCN), replacing the fully connected layers with more convolutional layers, has been a popular strategy and baseline for semantic segmentation [60, 174]. Long et al. [60] defined a novel architecture that combined semantic information from a deep, coarse layer with appearance information from a shallow, fine layer to produce accurate and detailed segmentations. DeepLab [174] proposed a similar FCN model, but also integrated the strength of conditional random fields (CRFs) into FCN for detailed boundary recovery. Instead of using CRFs as a post-processing step, Lin et al. [177] jointly trains the FCN and CRFs by efficient piecewise training. Likewise, the work in [178] converted the CRFs as a recurrent neural network (RNN), which can be plugged in as a part of FCN model.

(3) Weakly supervised annotations. Apart from the advancements in segmentation models, some works are focused on weakly supervised segmentation. Papandreou et al. [179] studied the more challenging segmentation with weakly annotated training data such as bounding boxes or image-level labels. Likewise,

the BoxSup method in [180] made use of bounding box annotations to estimate segmentation masks, which are used to update network iteratively. These works both showed excellent performance when combining a small number of pixel-level annotated images with a large number of bounding box annotated images.

We compare their results on PASCAL VOC 2012 val and test set in Table 2.7.

**Table 2.7:** Semantic segmentation results on PASCAL VOC 2012 val and test set

| Methods | Train | Val 2012 | Test 2012 |
|---|---|---|---|
| SDS [68] | VOC extra | 53.9 | 51.6 |
| CFM [175] | VOC extra | 60.9 | 61.8 |
| FCN-8s [60] | VOC extra | - | 62.2 |
| Hypercolumn [176] | VOC extra | 59 | 62.6 |
| DeepLab [174] | VOC extra | 63.7 | 66.4 |
| DeepLab-MSc-LargeFOV [174] | VOC extra | 68.7 | 71.6 |
| Piecewise-DCRFs [177] | VOC extra | 70.3 | 70.7 |
| CRF-RNN [178] | VOC extra | 69.6 | 72.0 |
| BoxSup [180] | VOC extra+COCO | 68.2 | 71.0 |
| Cross-Joint [179] | VOC extra+COCO | 71.7 | 73.9 |

### 2.3.5 Human Pose Estimation

Human pose estimation aims to estimate the localization of human joints from still images or image sequences, as shown in Figure 2.21.. It is very important for a wide range of potential applications, such as video surveillance, human behavior analysis, human-computer interaction (HCI), and is being extensively studied recently [181–191]. However, this task is also very challenging because of the great variation of human appearances, complicated backgrounds, as well as many other nuisance factors, such as illumination, viewpoint, scale, etc. In this part, we mainly summarize deep learning schemes to estimate the human articulation from still images, although these schemes could be incorporated with motion features to further boost their performance in videos [181–183].

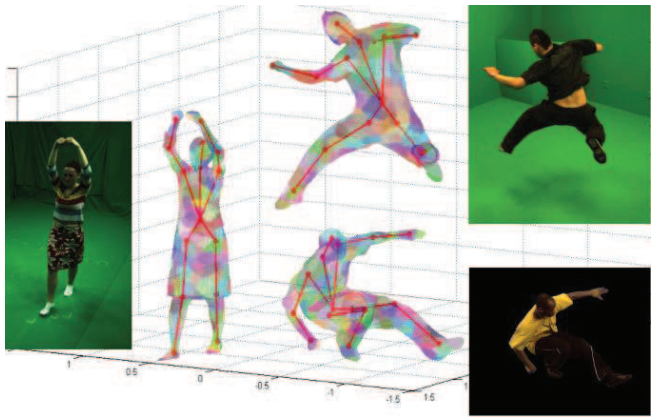


**Figure 2.21:** Human pose estimation [192].

Normally, human pose estimation involves multiple problems such as recognizing people in images, detecting and describing human body parts, and modeling their spatial configuration. Prior to deep learning, the best performing human pose estimation methods were based on body part detectors, i.e. detect and describe the human body part first, and then impose the contextual relations between local parts. One typical part-based approach is pictorial structures [193], which takes advantage of a tree model to capture the geometric relations between adjacent parts and has been developed by various well-known part-based methods [194–197].

As deep learning algorithms can learn high-level features which are more tolerant to the variations of nuisance factors, and have achieved success in various computer vision tasks, they have recently received significant attention from the research community.

We have summarized the performance of related deep learning algorithms on two commonly used datasets: Frames Labeled In Cinema (FLIC) [198] and Leeds Sports Pose (LSP) [199]. FLIC consists of 3987 training images and 1016 test images obtained from popular Hollywood movies, containing people in diverse poses, annotated with upper-body joint labels. LSP and its extension contains 11000 training and 1000 testing images of sports people gathered from Flickr with 14 full body joints annotated. There are two widely accepted evaluation metrics

for the evaluation: Percentage of Correct Parts (PCP) [200], which measures the rate of correct limb detection, and Percent of Detected Joints (PDJ) [198], which measures the rate of correct limb detection.

In the following, Table 2.8 illustrates the PDJ comparison of various deep learning methods on FLIC dataset, with a normalized distance of 0.05, and Table 2.9 lists out the PCP comparison on LSP dataset.

**Table 2.8:** The PDJ comparison on FLIC dataset

| PDJ(PCK) | Head | Shoulder | Elbow | Wrist |
|---|---|---|---|---|
| Jain et al. [186] | - | 42.6 | 24.1 | 22.3 |
| DeepPose [201] | - | - | 25.2 | 26.4 |
| Chen et al. [185] | - | - | 36.5 | 41.2 |
| DS-CNN [190] | - | - | 30.5 | 36.5 |
| Tompson et al. [187] | 90.7 | 70.4 | 50.2 | 55.4 |
| Tompson et al. [188] | 92.6 | 73 | 57.1 | 60.4 |

**Table 2.9:** The PCP comparison on LSP dataset

| | Torso | Head | U.arms | L.arms | U.legs | L.legs | Mean |
|---|---|---|---|---|---|---|---|
| Ouyang et al. [189] | 85.8 | 83.1 | 63.3 | 46.6 | 76.5 | 72.2 | 68.6 |
| DeepPose [201] | - | - | 56 | 38 | 77 | 71 | - |
| Chen et al. [185] | 92.7 | 87.8 | 69.2 | 55.4 | 82.9 | 77 | 75 |
| DS-CNN [190] | 98 | 85 | 80 | 63 | 90 | 88 | 84 |

In general, deep learning schemes in human pose estimation can be categorized according to the handling manner of input images: holistic processing or part-based processing.

The holistic processing methods tend to accomplish their task in a global manner, and do not explicitly define a model for each individual part and their spatial relationships. One typical model is called DeepPose proposed by Toshev et al. [201]. This model formulates the human pose estimation method as a joint regression problem and does not explicitly define the graphical model or part detectors for the human pose estimation. More specifically, it utilizes a two-layer architecture:

the first layer addresses the ambiguity between body parts in a holistic way and generates the initial pose estimation. The second layer refines the joint locations for the estimation. This model achieved advances on several challenging datasets. However, the holistic-based method suffers from inaccuracy in the high-precision region, since it is difficult to learn direct regression of complex pose vectors from images.

The part-based processing methods propose to detect the human body parts individually, followed with a graphic model to incorporate the spatial information. Instead of training the network using the whole image, Chen et al. [185] utilized the local part patches and background patches to train a DCNN, in order to learn conditional probabilities of the part presence and spatial relationships. By incorporating with graphic models, the algorithm gained promising performance. Moreover, Jain et al. [186] trained multiple smaller convnets to perform independent binary body-part classification, followed with a higher-level weak spatial model to remove strong outliers and to enforce global pose consistency. Similarly, Tompson et al. [187] designed multi-resolution ConvNet architectures to perform heat-map likelihood regression for each body part, followed with an implicit graphic model to further promote joint consistency. The model was further extended in [188], which argues that the pooling layers in the CNNs would limit spatial localization accuracy and try to recover the precision loss of the pooling process. They especially improve the method from [187] by adding a carefully designed Spatial Dropout layer, and present a novel network which reuses hidden-layer convolutional features to improve the precision of the spatial locality.

There are also approaches which suggesting combining both the local part appearance and the holistic view of the parts for more accurate human pose estimation. For example, Ouyang et al. [189] derived a multi-source deep model from a Deep Belief Net (DBN), which attempts to take advantage of three information sources of human articulation, i.e. mixture type, appearance score and deformation, and combine their high-level representations to learn holistic, high-order human body articulation patterns. On the other hand, Fan et al. [190] proposed a dual-source convolutional neutral network (DS-CNN) to integrate the holistic and partial view in the CNN framework. It takes part patches and body patches

as inputs to combine both local and contextual information for more accurate pose estimation.

As most of the schemes tend to design new feed-forward architectures, Carreira et al. [191] introduced a self-correcting model, called Iterative Error Feedback (IEF). This model can encompass rich structure in both input and output spaces by incorporating top-down feedback, and shows promising results.

## 2.4 Trends and Challenges

Along with the promising performance deep learning has achieved, the research literature has indicated several important challenges as well as the inherent trends, which are described next.

### 2.4.1 Theoretical Understanding

Although promising results in addressing computer vision tasks have been achieved by deep learning methods, the underlying theory is not well understood, and there is no clear understanding of which architectures should perform better than others. It is difficult to determine which structure, how many layers, or how many nodes in each layer are proper for a certain task, and it also need specific knowledge to choose sensible values such as the learning rate, the strength of the regularizer, etc. The design of the architecture has historically been determined on an ad-hoc basis. Chu et al. [202] proposed a theoretical method for determining the optimal number of feature maps. However, this theoretical method only worked for extremely small receptive fields. To better understand the behavior of the well-known CNN architectures, Zeiler et al. [22] developed a visualization technique that gave insight into the function of intermediate feature layers. By revealing the features in interpretable patterns, it brought further possibilities for better architecture designs. A similar visualization was also studied by Yu et al. [203].

Apart from visualizing the features, RCNN [44] attempted to discover the learning pattern of CNN. It tested the performance in a layer-by-layer pattern during the training process, and found that the convolutional layers can learn more general features and convey most of the CNN representational capacity, while the top fully-connected layers are domain-specific. In addition to analyzing the CNN features, Agrawal et al. [204] further investigated the effects of some commonly used strategies on CNN performance, such as fine-tuning and pre-training, and provided evidence-backed intuitions to apply CNN models to computer vision problems.

Despite the progress achieved in the theory of deep learning, there is significant room for better understanding in evolving and optimizing the CNN architectures toward improving desirable properties such as invariance and class discrimination.

### 2.4.2 Human-level Vision

Human vision has a remarkable proficiency in computer vision tasks, even in simple visual representations or under changes to geometric transformations, background variation, and occlusion. Human-level vision can refer to either bridging the semantic gap in terms of accuracy or in bringing new insights from studies of the human brain to be integrated into machine learning architectures. Compared with the traditional low-level features, a CNN mimics human brain structure and builds multi-layers activations for mid-level or high-level features. The study in [61] aimed to evaluate how much retrieval improvement can be achieved by developing deep learning techniques, and whether deep features are a desirable key to bridge the semantic gap in the long term. As seen in Figure 2.18, the image classification error on the ImageNet test set decreases 10%, from 15.3% [14] in 2012 to 4.82% [151] in 2015. This promising improvement verifies the efficiency of CNNs. In particular, the result in [151] has exceeded the accuracy of human raters. However, we cannot conclude that the representational performance of a CNN rivals that of the brain [205]. For example, it is easy to produce images

that are completely unrecognizable to humans, but one state-of-the-art CNN believes it to contain recognizable objects with 99.99% confidence [206]. This result highlights the difference between human vision and current CNN models, and raises questions about the generality of CNNs in computer vision. The study in [205] found that, like the IT cortex, recent CNNs could generate similar feature spaces for the same category, and distinct ones for images with different categories. This result indicates that CNNs may provide insight into understanding primate visual processing. In another study [207], the authors considered a novel approach for brain decoding for fMRI data by leveraging unlabeled data and multi-layer temporal CNNs, which learned multiple layers of temporal filters and trained powerful brain decoding models. Whether CNN models that rely on computational mechanisms are similar to the primate visual system is yet to be determined, but it has the potential for further improvements by mimicking and incorporating the primate visual system.

### 2.4.3 Training with limited data

Larger models demonstrate more potential capacity and have become the tendency of recent developments. However, the shortage of training data may limit the size and learning ability of such models, especially when it is expensive to obtain fully labeled data. How to overcome the need for enormous amounts of training data and how to train large networks effectively remains to be addressed.

Currently, there are two commonly used solutions to obtain more training data. The first solution is to generalize more training data from existing data based on various data augmentation schemes, such as scaling, rotating and cropping. On top of these, Wu et al. [57] further adopted color casting, vignetting and lens distortion techniques, which could produce much more converted training examples with broad coverage. The second solution is to collect more training data with weak learning algorithms. Recently, there has been a range of articles on learning visual concepts from image search engines [208, 209]. In order to scale up computer vision recognition systems, Zhou et al. [168] proposed the ConceptLearner

approach, which could automatically learn thousands of visual concept detectors from weakly labeled image collections. Besides that, to reduce laborious bounding box annotation costs for object detection, many weakly-supervised approaches have emerged with image-level object-presence labeling [210]. Nevertheless, it is promising to further develop techniques for generating or collecting more comprehensive training data, which could make the networks learn better features that are robust under various changes, such as geometric transformations, and occlusion.

### 2.4.4 Time complexity

The early CNNs were seen as a method that required a lot of computational resources and were not candidates for real-time applications. One of the trends is towards developing new architectures which allow running a CNN in real-time. The study in [59] conducted a series of experiments under constrained time cost, and proposed models that are fast for real-world applications, yet are competitive with existing CNN models. In addition, fixing the time complexity also helps to understand the impacts of factors such as depth, numbers of filters, filter sizes, etc. Another study [211] eliminated all the redundant computations in the forward and backward propagation in CNNs, which resulted in a speedup of over 1500 times. It has robust flexibility for various CNN models with different designs and structures, and reaches high efficiency because of its GPU implementation. Ren et al. [212] converted the key operators in deep CNNs to vectorized forms, so that high parallelism can be achieved given basic parallelized matrix-vector operators. They further provided a unified framework for both high-level and low-level vision applications.

### 2.4.5 More Powerful Models

As deep learning related algorithms have moved forward the-state-of-the-art results of various computer vision tasks by a large margin, it becomes more chal-

lenging to make progress on top of that. There might be several directions for more powerful models:

The first direction is to increase the generalization ability by increasing the size of the networks [24, 25]. Larger networks could normally bring higher quality performance, but care should be taken to address the issues this may cause, such as overfitting and the need for a lot of computational resources.

A second direction is to combine the information from multiple sources. Feature fusion has long been popular and appealing, and this fusion can be categorized in two types. 1) Combine the features of each layer in the network. Different layers may learn different features [44]. It is promising if we could develop an algorithm to make the features from each layer to be complementary. For example, DeepIndex [213] proposed to integrate multiple CNN features by multiple inverted indices, including different layers in one model or several layers from distinct models. 2) Combine the features of different types. We can obtain more comprehensive models by integrating with other type of features, such as SIFT. To improve the image retrieval performance, DeepEmbedding [214] used the SIFT features to build an inverted index structure, and extracted the CNN features from the local patches to enhance the matching strength.

A third direction towards more powerful models is to design more specific deep networks. Currently, almost all of the CNN-based schemes adopt a shared network for their predictions, which may not be distinctive enough. A promising direction is to train a more specific deep network, i.e. we should focus more on type of object we are interested in. The study in [43] has verified that object-level annotation is more useful than image-level annotation for object detection. This can be viewed as a kind of specific deep network which just focuses on the object rather than the whole image. Another possible solution is to train different networks for different categories. For instance, [215] built on the intuition that not all classes are equally difficult to distinguish from a true class label, and designed an initial coarse classifier CNN as well as several fine CNNs. By adopting a coarse-to-fine classification strategy, it achieves state-of-the-art performance on CIFAR100.

## 2.5 Conclusion

This chapter presents a comprehensive review of deep learning and develops a categorization scheme to analyze the existing deep learning literature. It divides the deep learning algorithms into four categories according to the basic model they derived from: Convolutional Neural Networks, Restricted Boltzmann Machines, Autoencoder and Sparse Coding. The state-of-the-art approaches of the four classes are discussed and analyzed in detail. For the applications in the computer vision domain, the chapter mainly reports the advancements of CNN based schemes, as it is the most extensively utilized and most suitable for images. Most notably, some recent articles have reported inspiring advances showing that some CNN-based algorithms have already exceeded the accuracy of human raters.

Despite the promising results reported so far, there is significant room for further advances. For example, the underlying theoretical foundation does not yet explain under what conditions they will perform well or outperform other approaches, and how to determine the optimal structure for a certain task. This chapter describes these challenges and summarizes the new trends in designing and training deep neural networks, along with several directions that may be further explored in the future.

# Chapter 3

# Convolutional Neural Networks Features: Principal Pyramidal Convolution

The features extracted from convolutional neural networks (CNNs) are able to capture the discriminative part of an image and have shown superior performance in visual recognition. Furthermore, it has been verified that the CNN activations trained from large and diverse datasets can act as generic features and be transferred to other visual recognition tasks. In this chapter, we aim to learn more from an image and present an effective method called Principal Pyramidal Convolution (PPC). The scheme first partitions the image into two levels, and extracts CNN activations for each sub-region along with the whole image, and then aggregates them together. The concatenated feature is later reduced to the standard dimension using Principal Component Analysis (PCA) algorithm, generating the refined CNN feature. When applied in image classification and retrieval tasks, the PPC feature consistently outperforms the conventional CNN feature, regardless of the network type where they derive from.

## 3.1 Introduction

Convolutional Neural Networks (CNN) have achieved breakthrough achievements in various visual recognition tasks and have been extensively studied in recent years [22, 44, 45, 213]. There are several brilliant properties for the CNN feature: 1)It is highly discriminative. Related research has analyzed the behavior of the intermediate layers of CNN and demonstrated that it can capture the most obvious features [22], thus could achieve considerably better results in a number of applications [22, 213]; 2) Unlike the hand-crafted features such as SIFT [1], HOG [3], the CNN feature is generated from end-to-end, which eliminates the human intervention; 3) it can be achieved efficiently. In contrast to the standard feedforward neural networks with similarly-sized layers, CNN has fewer connections and parameters, which reduces the time cost of the feature extraction; 4) it is transferrable. Some works [45, 62] have demonstrated that CNN features trained on large and diverse datasets, such as ImageNet [149] and Places [216], could be transferred to other visual recognition tasks, even there are substantial differences between the datasets.

Owing to those notable characters, our research focuses on reusing of the off-the-shelf CNN feature. But, instead of computing the CNN feature over the full image, we ask whether we could get more information from an image and achieve a refined version of the CNN feature?

An intuitive way to achieve more knowledge is to extract multiple CNN features from one image and organize them in a proper way. In recent years, there are a number of works attempt to extract multiple features from one image, either in region proposals [44] or sliding windows [170]. But most of those methods are used for object detection, not for the refinement of CNN features. Besides, the extraction of numerous features from overlapping regions is quite inefficient.

Related works have been done in the past [170, 217]. In the work by Gong et al. [170], they extract CNN activations at multiple scale levels, perform orderless VLAD pooling separately, and concatenate them together, forming a high dimensional feature vector which is more robust to the global deformations. Koskela et al. [217] splits one image into nine regions and averages their CNN activations,

concatenating with the activation of the entire image. The resulting spatial pyramid features are certificated to be more effective in scene recognition.

Different from previous works, we show that the concatenation of the CNN features from one image could also improve the performance, without further calculation or other time-consuming processes. To avoid increasing the complexity during the test phase and keep the key components at the meanwhile, we compress the dimension to the normal one (4096-D) using PCA scheme after the concatenation and get the refined feature: Principal Pyramidal Convolution (PPC).

The idea of concatenating features has ever been done in the literatures. The most representative one is the spatial pyramid matching (SPM) [135] algorithm, which concatenates the BOF vectors of the sub-regions as well as the whole image to import the global spatial information. SPM achieves a substantial improvement over the traditional BOF and has long been a key component in the competition-winning systems for visual recognition before the surge of CNN [31, 122].

In this chapter, the BOF vector of the SPM algorithm is replaced by the discriminative CNN feature. Therefore, besides preserving the discrimination of CNN, PPC also introduces some spatial information as well as preserving the most important components. In addition, the strategy is portable, experiments show that whichever network the CNN activations derive from, PPC strategy could consistently improve the performance.

## 3.2 Principal Pyramidal Convolution

Inspired by SPM, which extracts features at multiple levels and aggregates them together, we propose the Principal Pyramidal Convolution (PPC) method. It divides the image into two levels and generates the final feature for the image by concatenating and extracting principal components for the features at all resolutions. The basic idea is illustrated in Figure 3.1.

We extract CNN features from two scale levels. The first level corresponds to the full image, and the second level consists of $2 \times 2$ regions by equally partitioning

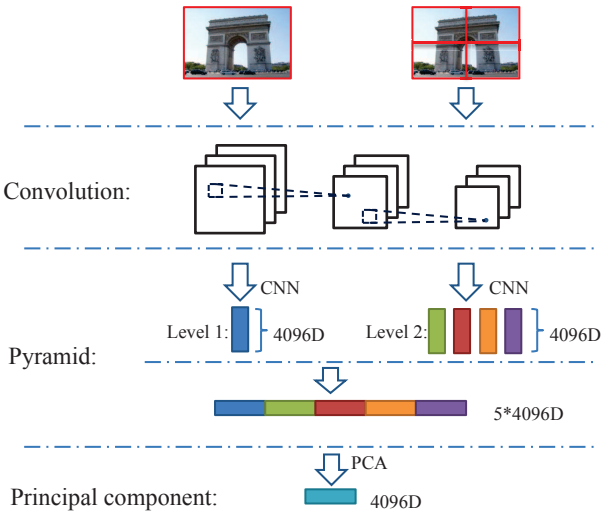



**Figure 3.1:** The procedure of PPC algorithm.

the full image. Therefore, we need to extract five CNN features for each image: $C_0, C_1, C_2, C_3, C_4$. Afterwards, we concatenate the five CNN features in an intuitive scheme: $C = [C_0, C_1, C_2, C_3, C_4]$. The resulting $C$ is a 5*4096-dimensional vector. The CNN activations are achieved using the Caffe implementation [218]. Here, we select the 4096-dimensional output of the seventh layer (i.e. the last fully-connected layer) and L2-normalize it as the baseline CNN feature.

To eliminate the increase of computational cost, we compress the resulting feature vector to 4096-D in the last step. For the dimension reduction, we utilize the well-known PCA method [219], which could reduce the dimensionality of a data set and retain as much as possible of the variation at the same time.

In addition, we also reduce the dimension to other various sizes and compare the performance between conventional CNN and PPC for different visual tasks, including the supervised image classification and unsupervised image retrieval.

## 3.3 Experiment

In this part, we make some comparisons between the conventional CNN feature and PPC feature on various image classification and image retrieval databases.

### 3.3.1 Datasets

We present the results on four widely used datasets: Caltech-101 [220], Scene15 [135], MIT Indoor67 database [221] and INRIA Holidays [222].

The details of the datasets are summarized in Table 3.1.

**Table 3.1:** Details of the datasets

| Datasets | Details |
|---|---|
| Caltech-101: | 102 categories and a total of 9144 images, the image number per category ranges from 31 to 800. We follow the procedure of [15] and randomly select 30 images per class for training and test on up to 50 images per class; |
| Scene15: | 4485 greyscale images assigned to 15 categories. Each category has 200 to 400 images. We use 100 images per class for training and the rest for testing; |
| Indoors67: | 67 categories and 15620 images in total. The standard training/test split consists of 80 images for training and 20 images for testing per class; |
| Holidays: | 1491 images corresponding to 500 image instances. Each instance has 2-3 images describing the same object or location. The images have been rectified to a natural orientation. 500 images of them are used as queries. |

In the databases described above, the first three datasets are used for image classification, on which we train linear SVM classifiers (s=0, t=0) to recognize the test images, using the LIBSVM tool [223]. The last dataset is a standard benchmark for image retrieval, and the accuracy is measured by the mean Average Precision (mAP) [224].

### 3.3.2 Comparisons on different networks

According to which database the CNN is trained on, the CNN features can be categorized into two types: ImageNet-CNN and Places-CNN. ImageNet-CNN is the most commonly used model which is trained on the well-known database: ImageNet [149]. This database contains 1000 categories with around 1.3 million images and most of the images are object-centric. Places-CNN is another model which is trained on the recently proposed Places database and is scene-

centric [216]. This database contains about 2.5 million images assigned to 205 scene categories.

In this chapter, we utilize the off-the-shelf CNN features of ImageNet and Places respectively, and compare the performance of CNN and PPC on the four datasets. The results are shown in Table 3.2.

**Table 3.2:** The classification accuracies of SPM and CNN, PPC on different networks

| Datasets | ImageNet-CNN | ImageNet-PPC | Places-CNN | Places-PPC | SPM [135] |
|---|---|---|---|---|---|
| Caltech-101 | 86.44% | **87.45%** | 61.07% | 67.41% | 64.6% |
| Scene15 | 84.49% | 86.4% | 89.11% | **89.88%** | 81.4% |
| Indoor67 | 59.18% | 64.4% | 72.16% | **73.36%** | - |
| Holidays | 73.95% | **74.9%** | 71.71% | 73.43% | - |

From the table, we can see that the improvements of PPC over CNN vary from about 1 percent to 6 percent, depending on the network and dataset. We can further conclude that: 1) the features generated from CNN are more distinctive than SIFT in image classification, and this inherent merit brings about the improvement of PPC in contrast to SPM. 2) The features derived from ImageNet-CNN are more discriminative in classifying objects, thus perform better on the Caltech-101 dataset. In contrast, the features achieved from the Places-CNN are better at classifying scenes, and accordingly perform better on Scene15 and MIT Indoor67 datasets. On one hand, choosing a suitable network (i.e. choose ImageNet-CNN for object recognition, or choose Places-CNN for scene recognition) could bring a significant improvement in the performance. As is shown by the experiment on Caltech-101 database, the advantage of ImageNet-CNN feature over the Places-CNN feature is more than 25 percent (86.44% and 61.07% respectively). On the other hand, choosing an unsuitable network could highlight the benefit of PPC over CNN. For instance, when we utilize Places-CNN features on the Caltech-101 database, the improvement of PPC over CNN is more than 6 percent, rising from 61.07% to 67.41%. Similarly, when the ImageNet-CNN features are tested on the Indoor67 dataset, the refinement of PPC over CNN could also be more than 5 percent (from 59.18% to 64.4%). But no matter which type of networks is applied

on the datasets, PPC features consistently outperform the holistic CNN features, demonstrating the effectiveness of the strategy.

For both the CNN and PPC algorithms on the MIT Indoor67 dataset, we visualize the distance between the features of the top performing categories in 3-dimensional space using the classic multidimensional scaling technique [225]. As is shown in Figure 3.2, the axes correspond to the coordinates in the 3-dimensional space and the categories are buffet, cloister, florist, inside bus.

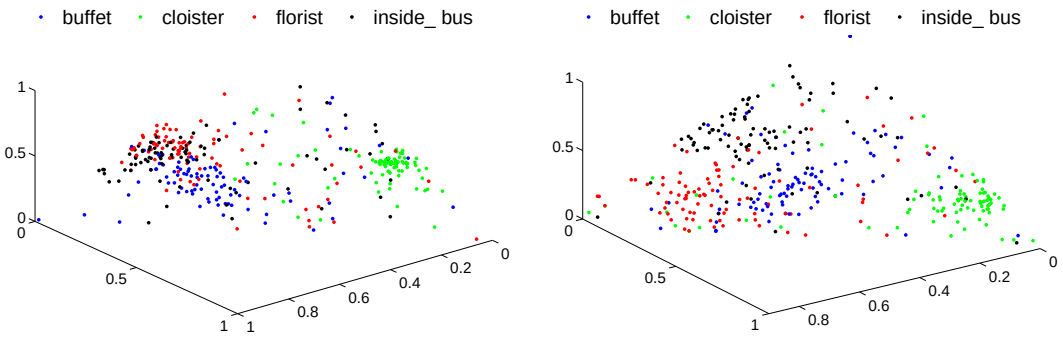



**Figure 3.2:** Top performing feature visualization of CNN(left) and PPC(right).

From the Figure 3.2, we can notice that the PPC features are more distinguishable than the holistic CNN features. The advantage is particularly evident on the comparisons between 'florist' and 'inside bus'.

For ImageNet-CNN model, we compare the accuracy of CNN and PPC on each category of Scene15 database, as is demonstrated in Figure 3.3. The x-axis details the categories and the y-axis corresponds to the accuracies of this category.

It can be observed that for most categories of Scene15 (ten of the fifteen categories), PPC performs better than CNN.

### 3.3.3 Comparisons on different dimensions

The improvement of PPC over CNN is not limited to 4096-D. To verify this, we further reduce the dimensionality to other sizes and compare the performance of

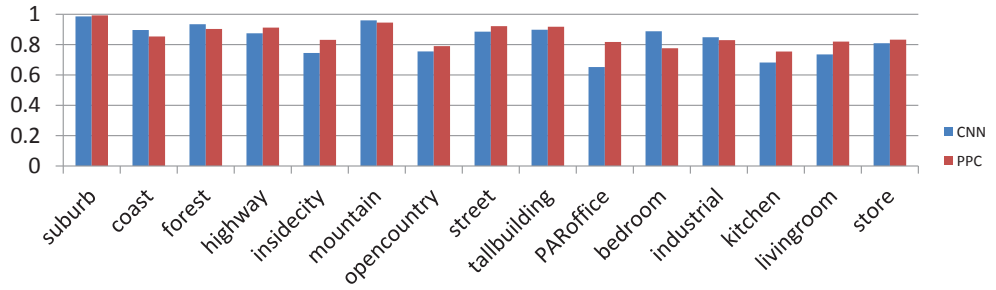



**Figure 3.3:** Accuracy of CNN and PPC on each category of Scene15.

PPC and CNN on different datasets, the results are shown in Figure 3.4.

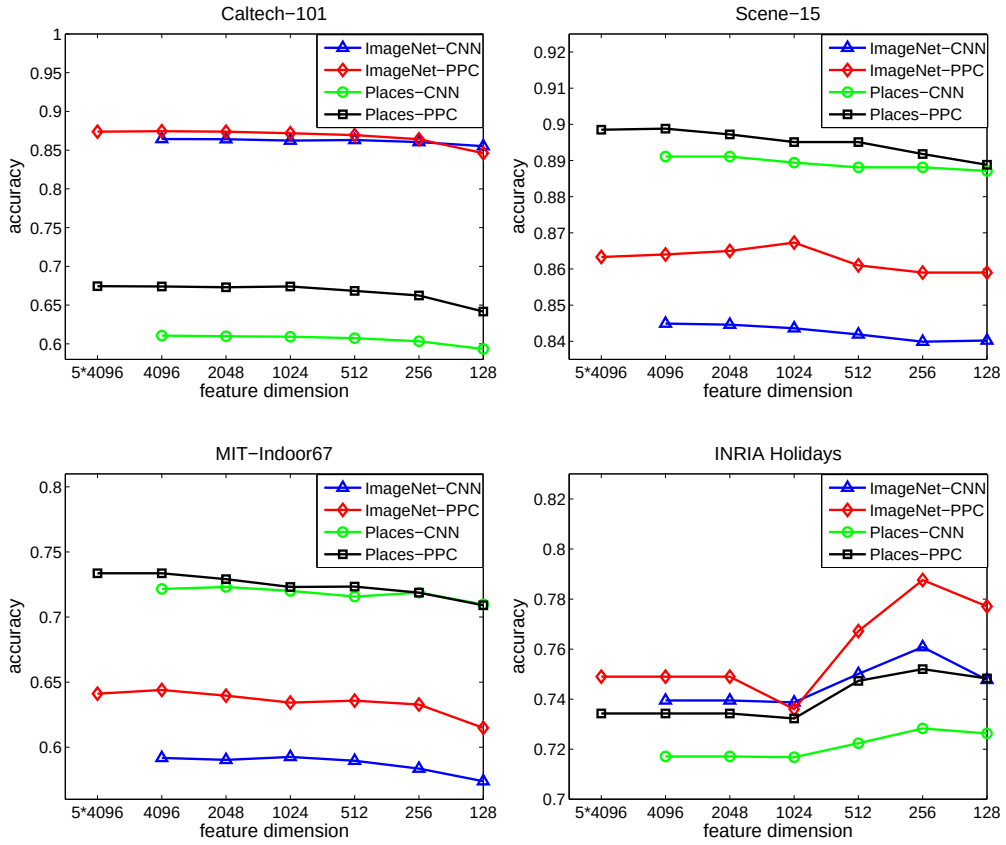



**Figure 3.4:** Comparisons of CNN and PPC on different dimensions

It is noticeable that the performance does not decrease even when the dimensionality of features are reduced to 128-D (most of the accuracies drift within

2 percent). On the contrary, the mAP of ImageNet-PPC on the INRIA Holidays dataset even rises to 78.76%, when the dimensionality is reduced to 256-D. This indicates that the discriminatory power of both CNN and PPC will not be greatly affected with the reduction of the dimensionality. Nevertheless, the performance of PPC is mostly better than that of CNN, indicating that PPC is more robustness than CNN.

## 3.4 Conclusion

CNN features have shown great promise in visual recognition. This chapter proposed the Principal Pyramidal Convolution (PPC) scheme, which aggregates the CNN features of the whole image as well as the sub-regions and then extracts the principal components. The representation from our strategy outperforms the conventional CNN feature without enlarging the feature dimensions. Furthermore, this work makes comparisons of CNN and PPC on different sizes and shows that the PPC frequently outperforms CNN.

# Chapter 4

# Bag of Surrogate Parts Feature for Visual Recognition

Convolutional Neural Networks (CNNs) have attracted significant attention in visual recognition. Several recent studies have shown that, in addition to the fully-connected layers, the features derived from the convolutional layers of CNNs can also achieve promising performance in image classification tasks. In this chapter, we propose a new feature from the convolutional layers, called Bag of Surrogate Parts (BoSP), and its spatial variant, Spatial-BoSP (S-BoSP). The main idea is, we assume the feature maps in the convolutional layers as surrogate parts, and densely sample and assign image regions to these surrogate parts by observing the activation values. Together with BoSP/S-BoSP, we further propose another two schemes to enhance the performance: scale pooling and global-part prediction. Scale pooling aims to handle the objects with different scales and deformations, and global-part prediction combines the predictions of global and part features. By conducting extensive experiments on generic object, fine-grained object and scene datasets, we find the proposed scheme can not only achieve superior performance to the fully-connected feature, but also produce competitive, or in some cases remarkably better performance than the state-of-the-art.

## 4.1 Introduction

Recently, convolutional neural networks (CNNs) have been widely used in visual recognition evaluations and achieved top tier performance on international benchmark datasets [226]. There have emerged some well-known CNN models, such as AlexNet [14], VGG [24], GoogLeNet [25] and ResNet [227]. It has been proved that these models, pretrained on ImageNet [149], can be employed as universal models and transferred to other visual recognition tasks [45, 228, 229].

In general, the CNN architecture consists of alternatively stacked convolutional layers and pooling layers, followed by several fully-connected layers. Initially, when utilizing the off-the-shelf CNN models, researchers tend to extract the image representation from the fully-connected layers, as they are reported to produce better results than the convolutional layers [22, 204]. However, compared with the fully-connected layers, there are several inherent advantages of the convolutional layers [230]. First, the activations of the convolutional layers contain more spatial information, because each spatial unit on the convolutional feature maps corresponds to one receptive field on the input image. Second, the convolutional features can be extracted from an image of any size and aspect ratio. Third, it has been demonstrated that the convolutional layers contain rich semantic information [231]. Owing to these promising advantages, many recent studies [230–235] have shifted to fully exploit the benefits of the convolutional layers.

A typical usage of the convolutional layers is to encode the convolutional features with the Bag-of-Words (BoW) variants, such as VLAD [5] and Fisher Vector [6]. This pipeline can not only preserve the high discrimination of the CNN activations, but also utilize the 'bag' conception to improve the invariance property to scale changes, location changes and occlusions. In this work, we also intend to generate features within the BoW framework, and accordingly propose a new feature, called Bag of Surrogate Parts (BoSP). The essential idea is: we assume the feature maps in the convolutional layers as surrogate parts, and define the activation values on the feature maps as assignment strengths for these surrogate parts. As each spatial unit on the feature maps corresponds to one image local region, the one-by-one processing of these spatial units acts like densely sampling

and assigning image regions. The final feature is generated by summing up the assignment strengths of different regions on the surrogate parts.

In comparison with prior research [70, 170, 230, 233, 235, 236] which also attempted to incorporate BoW and CNN, BoSP has several differences: First, BoSP does not need to generate the visual codebook, since the surrogate parts have already been inherently determined by the structure, i.e. the feature maps. This eliminates the time-consuming and sensitive process of visual dictionary learning. Second, in contrast to the features encoded by the variants of BoW [70, 230], BoSP is relatively in low dimension, making it advantageous in processing large scale datasets. Third, the surrogate parts are more semantically meaningful than the statistically clustered visual words. In Figure 4.1, we choose two images from SUN397 [237] and Indoor67 [221], and overlay some feature maps on the original images for visualization. As can be seen, the activated regions of the sampled feature maps indicate some semantically meaningful regions. For example, the activated region in top-left corner and top-right corner correspond to the 'table' and 'bed', respectively. A similar finding has also been presented in [231].

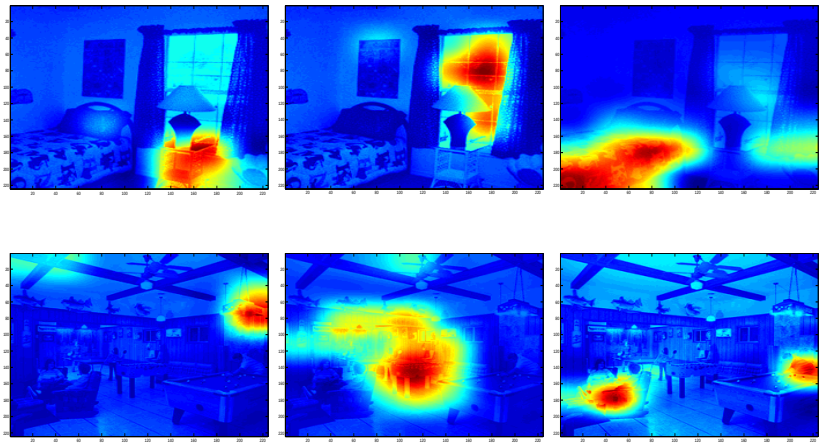


**Figure 4.1:** The visualization of the feature maps extracted from the last pooling layer of VGG [24].

Along with BoSP, there are three other contributions: (1) To incorporate more spatial information, we propose Spatial-BoSP (S-BoSP), by dividing the input image into several regions and concatenating the BoSP inside each region. (2)

To deal with the objects of different sizes and deformations, we develop a scale pooling scheme for the assignment of the spatial units, which could improve the performance considerably without enlarging the feature dimension and does not introduce much extra computational cost. (3) To make a more comprehensive prediction, we propose to multiply the global and part predictions, which constantly demonstrates better results than the individuals.

## 4.2 Related Work

**Part-based representation.** Parts can be used as mid-level visual elements to promote the object recognition process, and part-based approaches for object recognition have recently received much interest. Generally, the part-based methods can be viewed as a two-stage problem [238]. First, discover a collection of informative parts, and then train classifiers to detect the response of these parts. For example, Singh et al. [239] proposed an unsupervised method to find mid-level parts, by iteratively clustering HoG features and training classifiers. Juneja et al. [240] utilized the image-level labels to find the most discriminative parts based on entropy-rank curves, and employed BoW-based model to encode the features. Along with the promising achievements, the two-stage approaches also suffer from one possible drawback: the learned parts are not guarantied to be optimal for the classification task. As a consequence, several works [238, 241] suggested to jointly learn the parts and category models. On the other hand, Liu et al. [231] developed a scheme that did not explicitly define and detect the parts, but took the feature maps as the indicator maps of the parts, and concatenated the local features of parts as the image-level feature. This approach delivered encouraging results with a fraction of computational cost. Our method can be viewed as the combination of [231] and [240], as we do not explicitly define the parts either, and utilize the Bag of Parts model in [240] to represent the surrogate parts in [231].

**BoW-based schemes with deep CNN feature.** BoW-based methods have been widely used in previous researches, and achieved state-of-the-art perfor-

mance in various computer vision systems. In recent years, several studies [70, 170] attempted to introduce the deep CNN feature into the well-known BoW framework, especially its variants, such as VLAD [5] and Fisher Vector (FV) [6]. For instance, Gong et al. [170] extracted multiple fully-connected activations from three scales, and encoded them with VLAD scheme. Similarly, Yoo et al. [70] also extracted multi-scale top layer activations, but encoded them using the Fisher kernel. Additionally, Wu et al. [242] argued that the performances of FV and VLAD might fluctuate when different instant vectors were used, and proposed a more robust D3 (discriminative distribution distance) method. Although the aforementioned approaches achieved encouraging performance on various datasets, they all evolved a sensitive and computationally expensive process, i.e. learning the feature codebook. In comparison, the BoSP/S-BoSP are inherent features of the architecture which can be generated without manual tuning. This avoids the sensitive and time-consuming process of dictionary learning.

**Typical usage of convolutional features.** The convolutional features can be generally leveraged in two approaches. In the first approach, researchers encode the convolutional features with the variants of BoW scheme, such as VLAD and FV. For instance, Ng et al. [235] employed VLAD to encode the convolutional features and demonstrated that the intermediate layers could deliver better results for the image retrieval task than the top layers. In contrast, Cimpoi et al. [233] and Wei et al. [230] utilized FV to encode the descriptors from the intermediate layers, and also achieved promising performance. In the second approach, researchers take advantage of the convolutional activations in a more straightforward way, by aggregating and compressing them into the final representation. For example, Babenko et al. [232] aggregated the convolutional features in a simple sum-pooling way, and achieved a substantial boost in the performance. Liu et al. [231], on the other hand, took the feature maps as the indicator maps of parts, and aggregated the local features of each surrogate part as the image-level representation. Our method can be seen as the combination of these two approaches, in which we regard the feature maps as surrogate parts, similar to [231], but we do not explicitly concatenate the features of these parts. Instead, we only aggregate their statistical strengths, which makes the feature dimension much lower.

## 4.3 Bag of Surrogate Parts Feature

In this section, we first describe our proposed BoSP feature, and then give the interpretation of the surrogate part.

### 4.3.1 Bag of Surrogate Parts Feature

Generally, the CNN structure consists of multiple layers. Given an image, it will pass through several convolutional and pooling layers and generate various feature maps. We name the activation values on the feature maps as spatial units. As shown in Figure 4.2, one image region corresponds to multiple spatial units located in the same position on different feature maps, and the spatial units in higher feature maps have larger receptive fields than the lower ones.

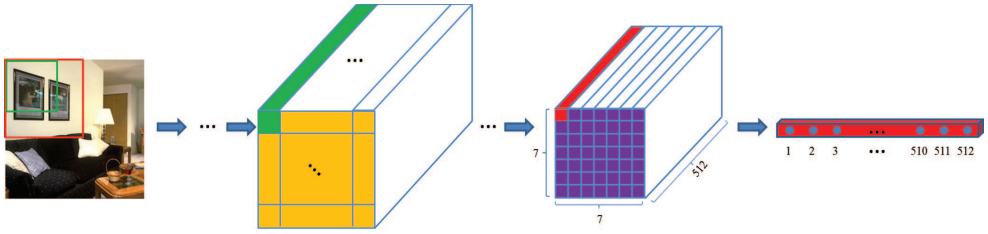



**Figure 4.2:** The illustration of the spatial unit. The local activations in green&red color are the responses of the green&red box surrounded image regions. For the VGG $\text{pool}_5$ layer, each image local region corresponds to 512 spatial units.

Intuitively, larger receptive field contains more semantic information. Therefore, we extract the BoSP feature from relatively higher layers (i.e. the 4th and the 5th pooling layer of VGG [24]. We simplified them as $\text{pool}_4$ and $\text{pool}_5$), as demonstrated in Figure 4.3. For the sake of clarity, we explain our method based on the $\text{pool}_5$ layer (For $\text{pool}_4$ layer, we first make an average pooling using $2 \times 2$ kernel with the stride 2, and then utilize the same operation with the $\text{pool}_5$ layer).

The specific procedure to extract BoSP is: we regard the feature maps as surrogate parts and assume that the activation values on the feature maps represent

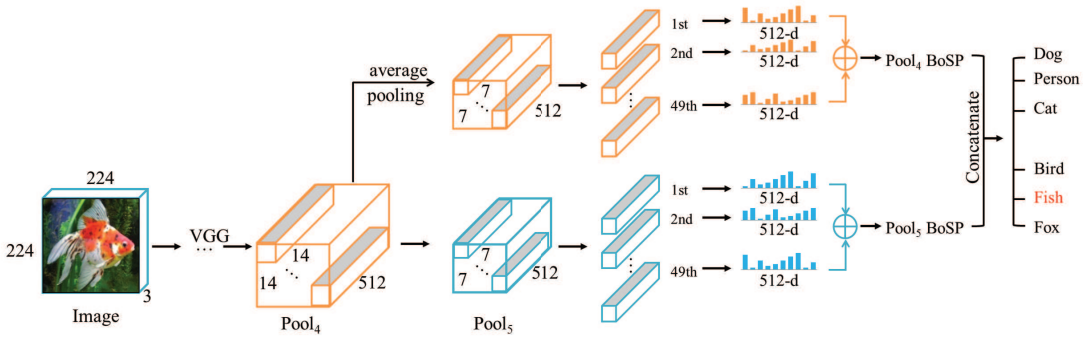



**Figure 4.3:** The framework of utilizing BoSP for image classification. We extract BoSP from the $\text{pool}_4$ and $\text{pool}_5$ layers of VGG. We first make an average pooling of $\text{pool}_4$ layer using $2 \times 2$ kernel with the stride 2, and then calculate the BoSP features of $\text{pool}_4$ and $\text{pool}_5$ layers. The final feature is the concatenation of $\text{pool}_4$ BoSP and $\text{pool}_5$ BoSP. $\bigoplus$ means element-wise addition of the vectors.

the assignment strengths for the surrogate parts. Therefore, given the architecture, the number of the surrogate parts is inherently determined, which equals the number of feature maps. For the spatial units on the feature maps, we can calculate their assignment strengths for the surrogate parts by observing the activation values. The one-by-one processing of these spatial units can be viewed as densely sampling and assigning spatial regions of the input image. Finally, we sum the assignment strengths for the surrogate parts and form a vector accordingly, i.e. BoSP, whose length equals the number of the feature maps.

More in detail, suppose there are $M$ feature maps and each feature map contains $n$ spatial units, then we have $M$ surrogate parts and can densely sample $n$ regions from the input image (for the $\text{pool}_5$ layer of VGG, $M = 512, n = 49$). The BoSP for this image can be writen as Eq 4.1:

$$BoSP = \sum_{i=1}^{n} [P_1^i, P_2^i, \cdots, P_j^i, \cdots, P_M^i] \tag{4.1}$$

$P_j^i$ represents the assignment strength of $i^{th}$ region on $j^{th}$ surrogate part.

To explicitly restrict the membership of the surrogate parts to [0,1], we normalize the local activations by diving the largest component of the vector, and take the

normalized activations as the assignment strengths, i.e.

$$P_j^i = \frac{A_j^i}{\max_j(A_j^i)} \tag{4.2}$$

$A_j^i$ means the $j^{th}$ element of the activation $A^i$ (It would always be non-negative since the activations are extracted after the ReLU layer).

To avoid using unreliable likelihood, we further adopt the idea of localized soft assignment [243] and only keep the assignment strengths with large values, and modify Eq 4.2 as Eq 4.3:

$$P_j^i = \begin{cases} 0 & \text{if } A_j^i < \text{mean}_j(A_j^i) \\ \frac{A_j^i}{\max_j(A_j^i)} & \text{if } A_j^i \geq \text{mean}_j(A_j^i) \end{cases} \tag{4.3}$$

The proposed BoSP has the following advantages: (1) It is efficient to be extracted. The feature is derived from the convolutional layers, which contain fewer parameters and need fewer computations. In practical, we only need to add up the larger normalized activation values of the feature maps. (2) It is less-specialized to be extracted. The dimension and the assignment strengths for the surrogate parts can be directly generated from the activation values, and there is no parameter for us to tune, which enhances its generality. (3) It is relatively low-dimensional. There are 512 feature maps in the $\text{pool}_5$ layer of VGG, so the dimension of BoSP from this layer is 512, much smaller than the schemes which need to concatenate convolutional features [230, 231, 234].

### 4.3.2 Interpretation of the Surrogate Part

In Figure 4.1, we have visualized that the feature maps can be viewed as surrogate parts. In this subsection, we try to give more insights into these surrogate parts by analyzing their influences on the categorization.

Similar to [244], we utilize regularized logistic regression method to make the prediction because it is faster and can help to evaluate the importance of the surrogate parts explicitly.

Logistic Regression assumes the probability for a binary classification satisfies:

$$\log \frac{p(y=1|x;\beta,w)}{p(y=-1|x;\beta,w)} = \beta + \sum_{j=1}^{d} w_j x_j \tag{4.4}$$

Where $p(y=1|x;\beta,w) + p(y=-1|x;\beta,w) = 1$, $x$ indicates the extracted BoSP feature vector, and $\beta, w$ are the parameters to learn.

From Eq 4.4, we can deduce that

$$p(y=1|x;\beta,w) = \frac{1}{1+\exp(-[\beta + \sum_{j=1}^{d} w_j x_j])} \tag{4.5}$$

By considering $\beta = w_0$ and $x_0 = 1$ , Eq 4.5 can be rewritten as:

$$p(y=1|x;\beta,w) = \frac{1}{1+\exp(-w^T x)} \tag{4.6}$$

The optimal parameter $w$ is obtained by optimizing the conditional log-likelihood function[245]:

$$\widehat{w} = \underset{w}{\operatorname{argmax}} \log \prod_i p(y_i|x_i;w) = \underset{w}{\operatorname{argmin}} \sum_i \log(1+\exp(-y_i w^T x_i)) \tag{4.7}$$

In this work, we use a L2-regularization term to restrict large values, as written in Eq 4.8:

$$\widehat{w} = \underset{w}{\operatorname{argmin}} \sum_i \log(1+\exp(-y_i w^T x_i)) + \lambda w^T w \tag{4.8}$$

Where $\lambda > 0$ is the regularization parameter.

Figure 4.4 demonstrates the learned weights for two categories (*Faces_ easy* and *beaver*) in the Caltech101 dataset, which correspond to the best-performing category (the accuracy of *Faces_ easy* is 100%) and worst-performing category (the accuracy of *beaver* is 62.5%) for the global prediction of BoSP. We can observe that, for different categories, the weights of the surrogate parts are different. A high positive value means the related surrogate part contributes a lot to the positive class, while a high negative value means the corresponding surrogate part

contributes a lot to the negative class. The surrogate classes with very small weights do not distinguish well between the positive and negative classes. For clarity, we denote the surrogate part which has the largest weight as *pos_part*, while the surrogate part with the smallest weight as *neg_part*.

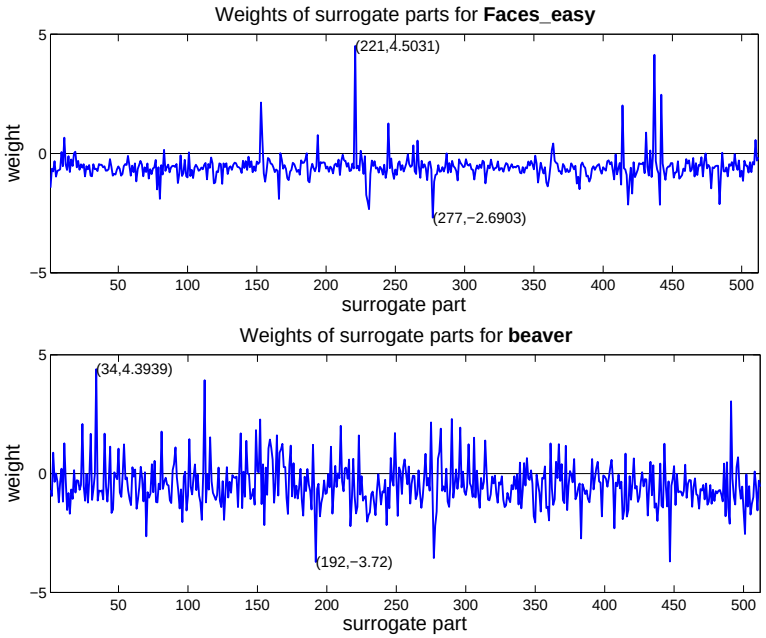



**Figure 4.4:** Learned weights for *Faces_easy* and *beaver* in Caltech101.

Ideally, given an input image, it should have large response in the *pos_part* of its groundtruth category, while have few response in the corresponding *neg_part*. To verify this, we select one positive image (i.e. correctly classified) from *Faces_easy* and one negative image (i.e. wrongly classified) from *beaver*. For these two images, we first draw the feature maps corresponding to the *pos_part* and *neg_part*, and then overlay the feature maps to original images for better visualization, as shown in Figure 4.5. The lightless on the feature maps represents the image's response on this surrogate part.

We can notice that, for the image of category *Faces_easy*, quite a lot of regions are assigned to its correct *pos_part*, and few regions are assigned to its *neg_part*, this contributes to its correct classification. In contrast, the wrongly classified image from category *beaver* contains quite a lot of regions assigned to its *neg_part*,

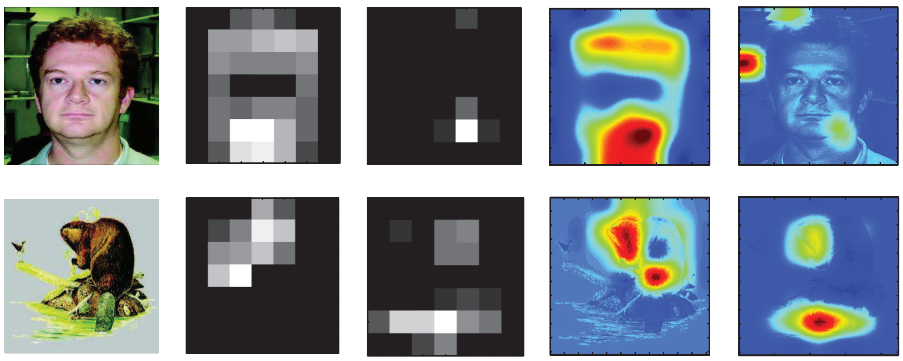



**Figure 4.5:** Visualization of the input images and the feature maps for the pos_part and neg_part. Pos_map and neg_map correspond to the feature maps which have the largest and smallest weight. Pos_overlay and neg_overlay demonstrate the activated regions when we overlay the corresponding feature maps to original images.

which leads to its mis-classification. Furthermore, when we overlay the feature maps to the original images, we found that these surrogate parts are semantically meaningful, and we could observe the image regions that affect the most for the classification. For example, the *pos_part* for the *beaver* image is located around the head area, while the *neg_part* corresponds to the 'floats', and the 'floats' contributes the most to its bad performance.

## 4.4 Enhancement schemes

In this section, we describe a spatial variant of BoSP, and propose two schemes to enhance the performance: scale pooling and global-part prediction.

### 4.4.1 Spatial BoSP

Motivated by the well-known spatial pyramid matching (SPM) method [135], we raise a spatial variant of BoSP, called S-BoSP. Specifically, we partition the image equally into multiple sub-regions (9 regions in 3 rows and 3 columns in

this chapter), calculate the BoSP inside each region and concatenate them into a single feature vector. Therefore, the dimension of S-BoSP is 9 times of BoSP. For simplicity, we conduct the partitioning process on the feature maps, rather than directly on the input images. As the size of the feature maps for the $\text{pool}_5$ layer of VGG is $7 \times 7$ , we need to divide the feature maps in an overlapping motion, i.e. some of the spatial units would exist in multiple sub-regions. The difference between BoSP and S-BoSP is illustrated in Figure 4.6.

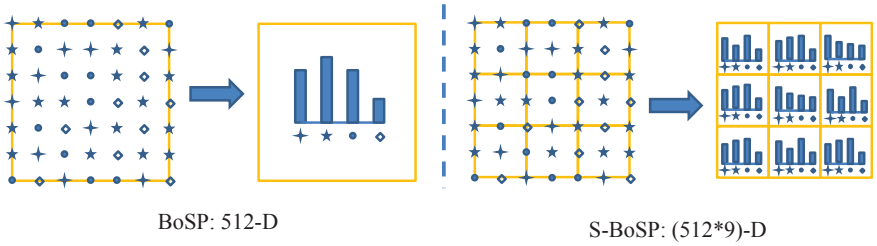



**Figure 4.6:** The illustration of BoSP feature and S-BoSP feature from the $\text{pool}_5$ layer of VGG (Different symbols represent different surrogate parts, and the histogram represents the assignment strength to these surrogate parts).

### 4.4.2 Scale Pooling

The BoSP/S-BoSP aforementioned only concern the spatial units at the finest level, and process them in a disjoint way, which means to sample and assign regions in input images with fixed size and position. However, the objects may appear in different positions, shapes and scales. The independent processing of the spatial units may capture different parts of the same object and have difficulties in classifying the objects of different scales. For example in Figure 4.7, the object 'water_lilly' from the two images appear in different scales. In this case, the fixed receptive region may capture different parts of the object. A $2 \times 2$ grid can capture most of the 'water_lilly' for the left image, while it can only capture some petals for the right image.

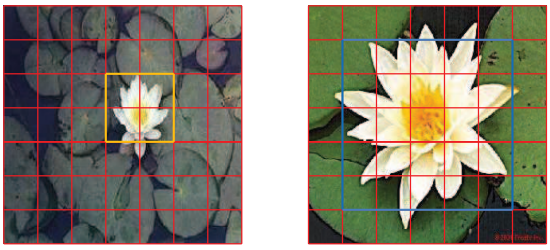


**Figure 4.7:** The demonstration of objects in different scale. For the left image, a $2 \times 2$ grid can capture most of the object, while the right image needs $6 \times 6$ grid to cover most of the object.

To address this problem, we proposed a scale pooling scheme. It can improve the assignment of objects with different scales and deformations by handling regions of different sizes and positions, together with the max pooling operation inside each region. The procedure of scale pooling is illustrated in Figure 4.8.

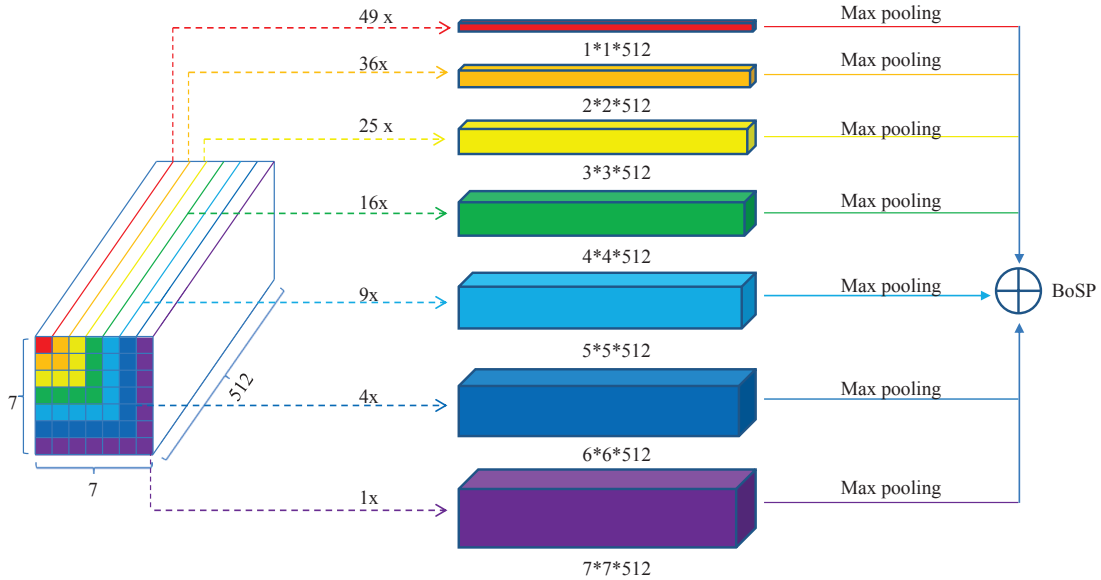



**Figure 4.8:** Pipeline of the scale pooling technique for BoSP. We can extract different number of features from 7 scales. For example, there are 49 red strips for the smallest scale, and only 1 purple strip for the largest scale. Next, we apply max pooling on the features inside each scale and calculate the BoSP individually, then add them up to form the final feature.

Specifically, we partition the activations from the $\text{pool}_5$ layer into 7 scales for BoSP. Under different scales, we derive spatial units of different numbers and different sizes. For clarity, we define the derived spatial units as coarse spatial units, and the coarse spatial units in scale 1 correspond to the original spatial units. Under scale $i$ ($i \in [1, 7]$), we can derive $(8-i)^2$ coarse spatial units, and each coarse unit contains $i^2$ original spatial units. Therefore, the total number of the coarse spatial units is $\sum_{i=1}^{7}(8-i)^2 = 140$. Next, we pool these coarse spatial units and compute their assignment strengths for the surrogate parts by employing Eq 4.3. In this work, we utilize the max pooling operation inside each coarse spatial unit since it has been proved to be superior for capturing invariance in image-like data [40]. Finally, we sum the assignment strengths of the coarse spatial units under different scales together to form the refined assignment strengths for the image. For S-BoSP, we utilize the scale pooling scheme inside each sub-region, and then concatenate the resulting features together.

To demonstrate the effectiveness of the scale pooling, we visualize the feature without/with scale pooling in Figure 4.9 using t-SNE technique [246], and we can see that the feature with scale pooling is more distinguishable.

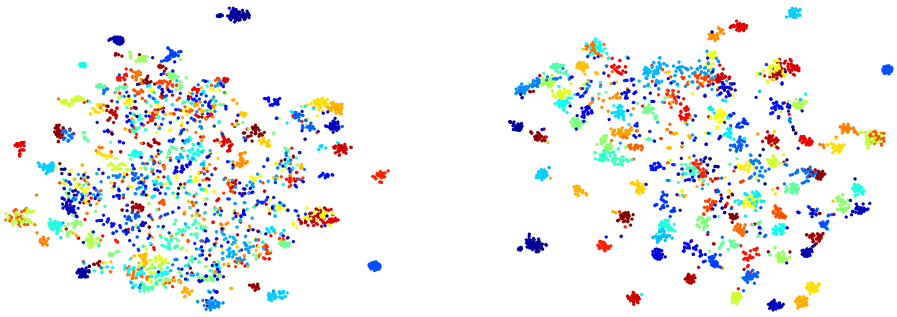


**Figure 4.9:** The visualization of feature without(left)/with(right) scale pooling for the Caltch101 dataset. Different symbol colors represent different categories.

Overall, the scale pooling scheme is proposed to make the assignment of receptive fields more comprehensive, by handling the image regions across different scales and positions. Benefiting from the max pooling operation, the scheme is robust

to object deformation inside each coarse spatial unit. In addition, scale pooling does not increase the feature dimension, nor does it have a significant effect on the computational efficiency.

### 4.4.3 Global-Part Prediction

For a given image, we first resize it to $224 \times 224$, and then extract its global feature by utilizing VGG. However, in many cases, extracting only one global feature from the input image is not discriminative enough, and many recent works [70, 170, 233, 235] proposed to extract multiple features from one single image, and generate a more comprehensive feature by integrating these features in a certain way. Without extra data, one common approach is to generate numerous sub-images from the input image, and average the sub-image features as augmented image feature. Although the augmented feature contains more information, it only considers individual parts of the input image, and fails to handle the input image entirely. To make a more comprehensive prediction, we propose to combine the predictions of the global feature and the augmented feature, as shown in Figure 4.10.

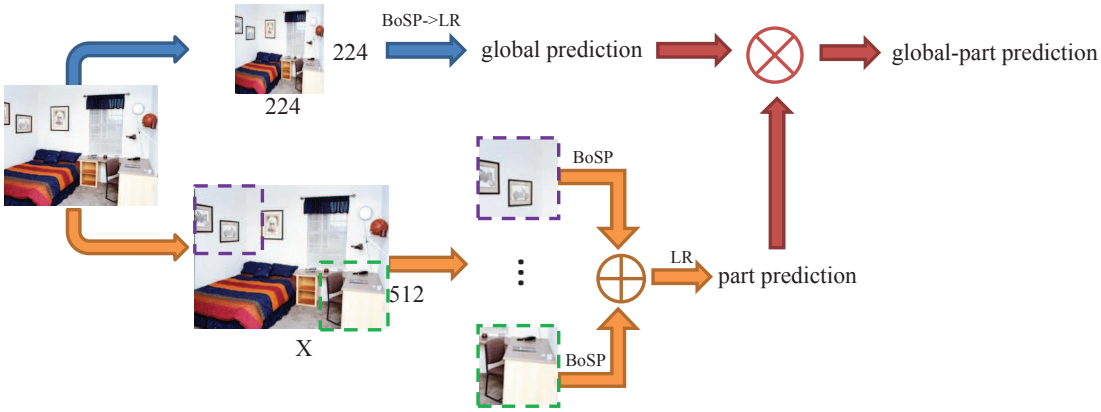



**Figure 4.10:** The illustration of global-part prediction. The global prediction is achieved by utilizing the global feature. The part prediction is achieved by averaging the parts' features. The global-part prediction is the product of the global prediction and the part prediction.

The specific procedure is: given an image, we first resize it to $224 \times 224$, and extract its global feature. This feature focuses more on the whole image, and we can compute the global prediction based on it, denoted as $Pre_{global}$; Next, we resize the image to make the smallest side equal to $S$ while keeping its ratio, and crop regions of $224 \times 224$ with the stride of 32 pixels. Thereby, we formulate several sub-images from the original image, and each sub-image may only contain part of the original object. The image feature is the average of the sub-images' features, as is the same with general approach. This feature focuses more about parts of the image, and we can make the part prediction based on it, denoted as $Pre_{part}$. The global-part prediction is the multiplication of the global prediction and the part prediction:

$$Pre_{global-part} = Pre_{global} \times Pre_{part} \tag{4.9}$$

As our feature is obtained from the convolutional layers, the input image could be of any size, and we do not need to explicitly crop sub-images. In practice, we only need to input the resized image once to extract the part features.

## 4.5 Experiments

To evaluate the performance of our method, we conduct a series of experiments on four datasets, Caltech101 [220], Oxford 102 Flowers (referred to as Oxford102) [247], MIT Indoor67 (referred to as Indoor67) [221] and SUN397 [237], which cover several popular topics in image classification, i.e. generic object classification, fine-grained object classification, and scene classification. Some of example images are shown in Figure 4.11. The details of the datasets are described below:

**Caltech101** consists of 9144 images in 102 object categories (101 object classes and a background class). The image number per category ranges from 31 to 800. For each category, we randomly select 30 images for training and test on up to 50 images. There are 44 'overlap' images of the Caltech101 dataset and ImageNet training data. We exclude these images from the test set.


**Figure 4.11:** The example images for the four datasets.

**Oxford102** has 102 flower categories and a total of 8189 images. Each category contains 40 to 258 images. The flowers appear under various scales, pose and illuminations. For each class we use 20 images for training and the rest for testing.

**Indoor67** contains 15620 images in 67 indoor categories. We use the standard train/test split provided in [221], which consists of 80 training and 20 test images per category.

**SUN397** is a large scale scene dataset from a collaboration between MIT and Brown University. It contains more than 100K images for 397 categories and is generally considered to be at a high difficulty level and very challenging. Each category has at least 100 images. The standard training/test splits are available from [237], and each split contains 50 training and 50 test images per category. We average the results of the 10 public splits as the final classification accuracy.

For all the experiments, we employ VGG Net-D [24] as the pre-trained CNN model to extract features. The model is implemented by the Caffe [218] package. For simplicity, pre-trained model weights are kept fixed without fine-tuning. All of the BoSP/S-BoSP features are L2 normalized before the experiment. The LR classifier is implemented by utilizing the open source library: LIBLINEAR [248].

For the global-part prediction, we resize the images of Caltech101 to let its smallest side $S = 256$, while $S = 512$ for Oxford102, Indoor67 and SUN397.

### 4.5.1 Analysis of our method

#### 4.5.1.1 Which classifier to use?

It has been demonstrated that the features delivered from CNN models are highly discriminative, and can be combined with classifiers to boost the image classification performance [24, 231]. However, most of these researches, if not all, adopt linear SVM (LSVM) classifier for their tasks, and ignore other classifiers which may corporate better with the features.

As we have demonstrated that each of the convolutional layers can be viewed as a surrogate part, we assume that it would be more reasonable if we can explicitly consider the differences among the feature maps, and propose to utilize the L2-regularized Logistic Regression (LR) classifier to handle our feature.

To verify our assumption, in this section, we compare the performance of three classifiers, i.e. LSVM, histogram intersection kernel SVM (HIKSVM) [249] and LR. All of the classifiers take the global BoSP feature derived from the $\text{pool}_5$ layer as the input. The results are shown in Table 4.1.

**Table 4.1:** The comparison of the accuracy and efficiency (training/test time) for different classifiers.

| | LSVM | HIKSVM | LR |
|---|---|---|---|
| Caltech101 | 86.68% | 87.01% | **88.28%** |
| time(s) | 19.17+16.27 | 35.16+34.39 | 6.46+0.13 |
| Oxford102 | 72.42% | 80.84% | **81.28%** |
| time(s) | 10.24+27.35 | 13.39+40.22 | 3.51+0.22 |
| Indoor67 | 67.46% | 69.40% | **69.48%** |
| time(s) | 43.92+13.14 | 107.84+26.13 | 13.49+0.05 |
| SUN397 | 51.23% | 53.31% | **53.68%** |
| time(s) | 587.96+954.74 | 1791.90+1969.10 | 320.62+2.01 |

In terms of accuracy, both the HIKSVM and LR perform better than the commonly used LSVM, demonstrating that we could have more options for the classifier, aside from LSVM. Particularly, the improvement of LR over LSVM is quite remarkable, from 1.6% to 8.86%. This verifies our assumption that, it is more reasonable to explicitly consider the differences among the feature maps.

In terms of efficiency, as HIKSVM needs to build non-linear kernels, it is the most computationally expensive, both for training and testing. Compared to these two classifier, LR is significantly faster due to its simple operations.

Owing to the advantages of LR, we utilized the LR classifier in all the following experiments. We further investigate the influence of the regularization parameter $\lambda$, by ranging the values from 5 to 50. As is shown in Figure 4.12, the influence on the accuracy is negligible when we change $\lambda$, and for fair comparisons, we set a fixed regularization term $\lambda = 20$.

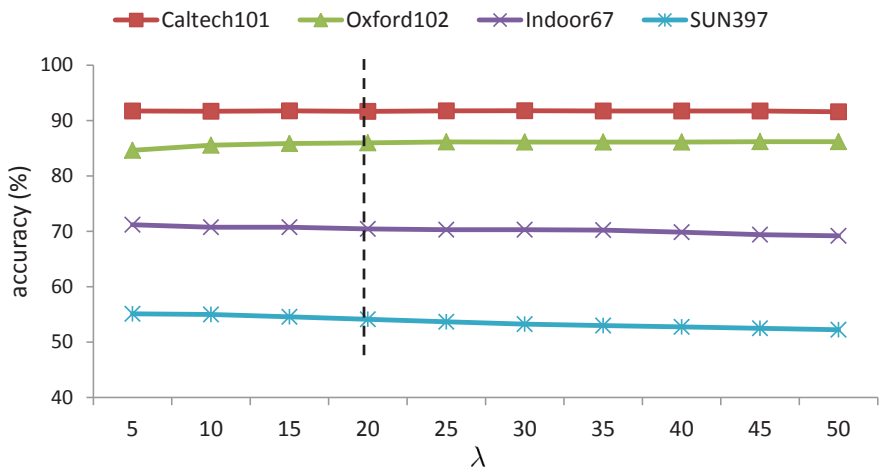



**Figure 4.12:** The performance of different regularization parameter values for training LR.

#### 4.5.1.2 The comparison of BoSP from different layers

The proposed BoSP is achieved from the convolutional layers, and we can formulate multiple BoSP features from different layers of the network. Intuitively, deeper layer activations would contain more semantic information compared to

shallower layer activations, thus should deliver better performance. To verify this, we evaluated the accuracy and efficiency of global BoSP from different layers of VGG on the Caltech101 dataset.

From Figure 4.13, we can see that, in terms of accuracy, the performance of BoSP would increase along with the layer depth, in which the feature derived from the $\text{pool}_5$ layer obtains the best result. This phenomenon confirms our assumption on the advantage of using deeper layers.

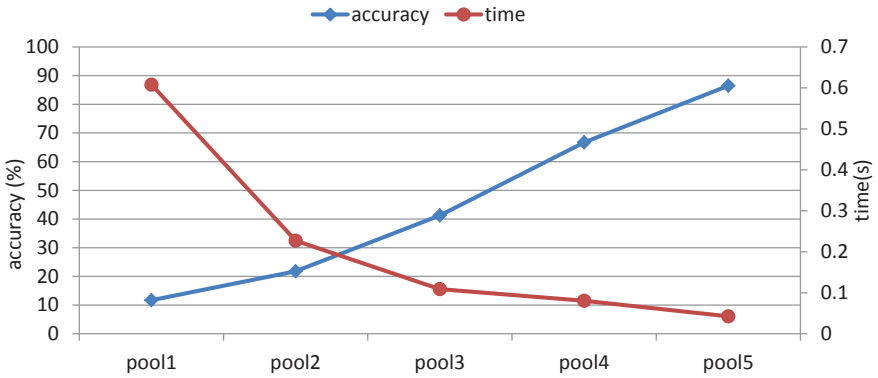



**Figure 4.13:** The performances of BoSP for different layers of VGG. In VGG, there are 5 major convolutional layers. We take the last sub-layer (i.e. the pooling layer) as the representative of the major layer.

As for the efficiency, it also improves with the layer depth, since the deeper feature maps are smaller than shallower ones, we need to assign fewer spatial units for the deeper layers. To combine the visual contents from different levels, we propose to extract features from $\text{pool}_4$ and $\text{pool}_5$ layers (Although concatenating more layers may further improve the performance, it would increase the feature dimension, thus is not good for large-scale data processing). To accelerate the feature extraction process from the $\text{pool}_4$ layer, we first make an average pooling using $2 \times 2$ kernel with the stride 2. In this way, the resulting feature maps from $\text{pool}_4$ layer share the same size with those from $\text{pool}_5$ layer.

#### 4.5.1.3 Evaluation of the Scale Pooling

In this part, we first evaluate the benefits brought by the proposed scale pooling scheme, and then compare the BoSP/S-BoSP with the commonly used CNN feature. All the features are extracted after resizing the images to $224 \times 224$.

Figure 4.14 reveals the merit of scale pooling on BoSP. For all the four datasets, the BoSP extracted with scale pooling outperforms the corresponding BoSP without it, and the advantage can be very large. For instance, for the Caltech101 dataset, scale pooling increases the $\text{pool}_4$ BoSP, $\text{pool}_5$ BoSP and concatenated BoSP by 10.12%, 3.37% and 3.94% respectively. Moreover, scale pooling would not enlarge the feature dimension, which demonstrates its great potential.

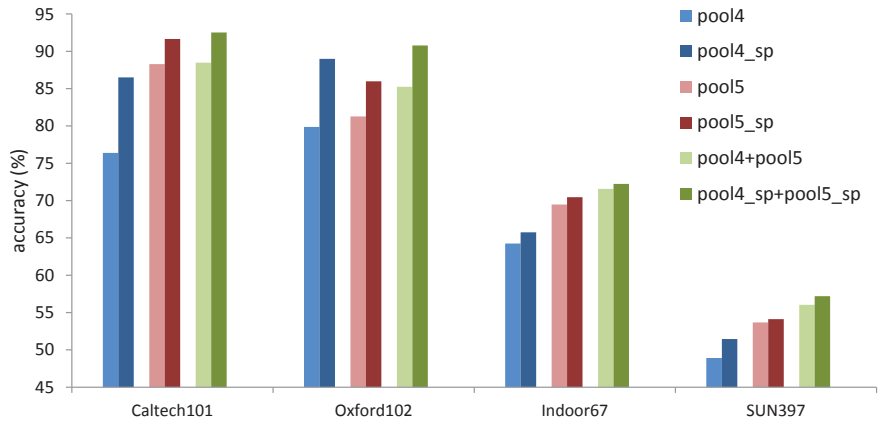



**Figure 4.14:** The comparison between BoSP with and without using scale pooling (The features using scaling pooling have the suffix: '_sp', and '+' means to concatenate features).

We further compare the proposed BoSP/S-BoSP with commonly used CNN feature (i.e. the activation from the last fully connected layer) in Table 4.2. As we can see, the BoSP/S-BoSP from the $\text{pool}_5$ layer could already achieve remarkably better performance than the CNN feature. After incorporating the BoSP from the $\text{pool}_4$ layer, the advantage become even more obvious. Take the Oxford102 as an example, the $\text{pool}_5$ BoSP brings 5.38% accuracy increase over the CNN feature, from 80.60% to 85.98%, while comes in much lower dimension (512 v.s.

4096). After concatenating the $pool_4$ BoSP, the improvement comes to 10.18% (from 80.60% to 90.78%).

Although the scale pooling is proposed to handle the objects of different scale for the BoSP/S-BoSP, technically, it can also be employed to the average pooling (AP). For a fair comparison, we list the AP with scale pooling in Table 4.2. We can see that BoSP/S-BoSP can achieve overall better performance than AP, especially for the concatenated features. A more detailed description of the relationship between BoSP and AP is presented in the discussion section.

**Table 4.2:** The comparison of AP/BoSP/S-BoSP with scale pooling and the CNN feature extracted from the last fully-connected layer.

| | Dim | Caltech101 | Oxford102 | Indoor67 | SUN397 |
|---|---|---|---|---|---|
| CNN | 4096 | 89.22% | 80.60% | 68.06% | 53.26% |
| $pool_5$, with scale pooling: | | | | | |
| AP | 512 | 91.62% | 85.82% | 69.63% | 53.93% |
| BoSP | 512 | 91.65% | 85.98% | 70.45% | 54.12% |
| S-BoSP | 4608 | 93.99% | 85.54% | 71.19% | 55.42% |
| [$pool_4$ , $pool_5$], with scale pooling: | | | | | |
| AP | 1024 | 92.49% | 90.4% | 70.97% | 56.04% |
| BoSP | 1024 | 92.55% | **90.78%** | 72.24% | 57.19% |
| S-BoSP | 9216 | **94.09%** | 89.92% | **73.21%** | **58.20%** |

From Table 4.2, we can also conclude another two findings: (1) Compared with using the $pool_5$ BoSP/S-BoSP individually, concatenating the $pool_4$ feature would double the feature dimension, but is always beneficial to the accuracy. Remarkably, the increases on Oxford102 and SUN397 are about 5% and 3%, respectively. (2) Except for Oxford102, S-BoSP performs better than the corresponding BoSP on the other three datasets, which demonstrates the effectiveness of the spatial scheme.

#### 4.5.1.4 Evaluation of the global-part prediction

In this subsection, we evaluate the proposed global-part prediction on both the BoSP and S-BoSP features. From the results in Table 4.3, it is obvious that the part prediction performs better than the global prediction, demonstrating that extracting multiple features in a single image is useful. Furthermore, we can also observe the advantage of global-part prediction over the global/part prediction: regardless of the differences between the global prediction and the part prediction, it is always beneficial to combine them by multiplication. Notably, the improvement on the predictions of SUN397 can be about 3%.

**Table 4.3:** The comparison of the different predictions on BoSP and S-BoSP. $\text{Pre}_{global}$: global prediction; $\text{Pre}_{part}$: part prediction; $\text{Pre}_{g-p}$: global-part prediction. The features are the concatenated features from $\text{pool}_4$ and $\text{pool}_5$ layer.

| | Caltech101 | | Oxford102 | | Indoor67 | | SUN397 | |
|---|---|---|---|---|---|---|---|---|
| | BoSP | S-BoSP | BoSP | S-BoSP | BoSP | S-BoSP | BoSP | S-BoSP |
| $\text{Pre}_{global}$ | 92.52% | 94.09% | 90.78% | 89.92% | 72.24% | 73.21% | 57.19% | 58.20% |
| $\text{Pre}_{part}$ | 92.62% | 94.12% | 93.54% | 92.94% | 77.46% | 77.31% | 60.48% | 60.97% |
| $\text{Pre}_{g-p}$ | 93.02% | **94.92%** | **94.02%** | 93.10% | **78.21%** | 78.13% | 63.21% | **63.79%** |

### 4.5.2 Comparison with the state-of-the-art

In Table 4.4, we list the comparison between our scheme and the published state-of-the-art schemes on the four datasets. All of the features are extracted based on the VGG network. We do not list the methods which employed additional information to improve classification, such as utilizing the part annotations for Oxford102, or using the large-scale scene-specific Places2 dataset[250] for Indoor67/SUN397.

We observe that, for the Caltech101 dataset, our scheme obtains the top accuracy. Notably, our BoSP feature from VGG Net-D achieves slightly better than the published result of VGG [24], which is 92.7%. However, their result is obtained by concatenating the fully-connected features from two models ( VGG Net-D & VGG Net-E) and three scales ($S = 256, 384, 512$), making the dimension of feature

**Table 4.4:** The comparison with the state-of-the-art (All of the methods are based on the VGG Network. Best results are in bold face).

| Method | Caltech101 | Oxford102 | Indoor67 | SUN397 |
|---|---|---|---|---|
| VGG [24] | 92.7% | - | - | - |
| ONE [251] | - | 86.82% | 70.13% | 54.87% |
| CrossLayer-OConv [234] | - | - | 74.4% | - |
| CrossLayer-AConv [234] | - | - | 78.2% | - |
| Deep19-DAG [252] | - | - | 77.5% | 56.2% |
| FV-CNN [233, 236] | - | - | **81%** | - |
| NML [253] | - | 84.3% | - | - |
| BoE [254] | - | - | 77.63% | - |
| D3(K=8) [242] | 93.80% | - | 77.76% | 60.22% |
| Best Single [255] | - | - | 76.42% | 59.71% |
| Dual [255] | - | - | 79.04% | 61.07% |
| Bayesian LS-SVM [256] | 93.3% | 91.5% | 77.8% | 56.1% |
| SCDA [257] | - | 92.1% | - | - |
| BoSP ($\text{Pre}_{global-part}$) | 93.02% | **94.02%** | 78.21% | 63.21% |
| S-BoSP ($\text{Pre}_{global-part}$) | **94.92%** | 93.10% | 78.13% | **63.79%** |

much larger than ours (12288 v.s. 1024). The proposed S-BoSP with global-part prediction further improves the state-of-the-art from 93.80% to 94.92%.

For the fine-grained Oxford102 dataset, the S-BoSP achieves inferior performance than BoSP, suggesting that the spatial scheme does not work for this dataset. We suspect this is because the small parts of the fine-grained objects are similar in appearance and do not distinguish well. Therefore, it is better to process the input as a whole image, without partitioning. Nevertheless, both BoSP and S-BoSP get better results than the state-of-the-art. Particularly, BoSP achieves considerable improvement over the previous best result, from 92.1% to 94.02%, with a dimension of 1024, demonstrating the effectiveness of our scheme.

For the Indoor67 dataset, the BoSP delivers competitive performance with the state-of-the-art (FV-CNN). In contrast to FV-CNN, BoSP has a much smaller (1K vs 64K) dimension, which can be a significant advantage in many situations. Compared with another recent work [255], our method achieves better

performance than its best single network, only slightly worse than its best dual architectures. However, the dual architectures can only be obtained after extensive comparisons, and it is not clear which two networks shall we choose before the experiment.

It is worth noting that [234] also takes the feature maps as indicator maps of surrogate parts, but it explicitly constructs and concatenates the features of each surrogate part to formulate the image feature. As a consequence, the resulting feature would be quite high dimensional, i.e. 262144-D for CrossLayer-OConv and 200000-D for CrossLayer-AConv. In contrast, we do not explicitly construct the surrogate part features and only focus on the assignment strength for these surrogate parts, making the feature dimension much lower. Nevertheless, we still get better performance than the CrossLayer-OConv, and competitive performance with the CrossLayer-AConv.

For the large scale, general scene classification on SUN397 dataset, our method obtains remarkably better performance than the current best result, improving the state-of-the-art from 61.07% to 63.79%.

## 4.6 Discussion

In this work, we regard the feature maps in convolutional layers as surrogate parts, and propose to utilize Eq 4.3 as the soft assignment for these surrogate parts. Under this assumption, we can also take advantage of other assignment schemes. For example, we have done some additional experiments to test the traditional soft assignment coding. The definition of traditional soft-assignment coding [258] is:

$$U_j^i = \frac{\exp(-\beta||D_j^i||)}{\sum_{k=1}^{n}\exp(-\beta||D_k^i||)} \tag{4.10}$$

Where $U_j^i$ denotes the membership of the $ith$ local feature to the $jth$ visual word, and $||D_j^i||$ is the distance between them. $\beta$ is the smoothing factor controlling the softness of the assignment.

In our scheme, the activation value has opposite assignment meaning with the distance, as a larger $A_j^i$ means the $ith$ local feature is more like to be assigned to the $jth$ surrogate part. Therefore, we modify the standard soft assignment as:

$$U_j^i = 1 - \exp(-\beta A_j^i) \tag{4.11}$$

Note that we do not explicitly constrain $\sum_{k=1}^{n} U_k^i = 1$ since there are many cases in which $A^i = 0$.

For the $\text{pool}_5$ layer of VGG, when we set $\beta = 0.02$, we can get similar results with this work, as shown in Table 4.5. This further verifies the reasonableness of our assumption to regard the feature maps as surrogate parts, and demonstrates the extensibility of our scheme.

**Table 4.5:** The comparison of BoSP with different soft assignment schemes. The BoSP with superscript $*$ means that we use Eq 4.11 as the soft assignment for the surrogate parts.

| | Caltech101 | Oxford102 | Indoor67 | SUN397 |
|---|---|---|---|---|
| BoSP | 91.65% | 85.98% | 70.45% | 54.12% |
| BoSP* | 91.55% | 86.31% | 70.45% | 54.29% |

**Comparison to average pooling:** Operationally, the straightforward assignment Eq 4.3 is a special case of average pooling. When we do not normalize the local activations and do not make them sparse, the BoSP would evolve to AP. However, conceptually, BoSP and AP have different views of the feature maps. AP processes the feature maps individually by averaging the values on each feature map, while BoSP handles the image local regions individually by normalizing the local activations. When we utilize other soft assignment schemes, e.g. Eq 4.11, their fundamental differences would become clearer.

## 4.7 Conclusion and Future Work

We proposed a new feature from the convolutional layers of VGG, which is highly discriminative and can be efficiently extracted. Along with the feature, we further introduced another three schemes to enhance the performance: spatial aggregation, scale pooling and global-part prediction. In addition, we also explored the semantic meaning of the surrogate parts and combined the BoSP feature from different layers. Our experiments in several popular classification tasks demonstrated the success of our scheme.

In the future, we would extend our work in three possible directions: (1) We would extract the feature from more advanced network (e.g. Res [227]) for further improvement; (2) We intend to directly utilize the scale pooling scheme for training the deep network; (3) We would employ our proposed feature for different applications, such as object detection and image retrieval.

# Chapter 5

# CNN-RNN: A Large-scale Hierarchical Image Classification Framework

Objects are often organized in a semantic hierarchy of categories, where fine-level categories are grouped into coarse-level categories according to their semantic relations. While previous works usually only classify objects into the leaf categories, we argue that generating hierarchical labels can actually describe how the leaf categories evolved from higher level coarse-grained categories, thus can provide a better understanding of the objects. In this chapter, we propose to utilize the CNN-RNN framework to address the hierarchical image classification task. CNN allows us to obtain discriminative features for the input images, and RNN enables us to jointly optimize the classification of coarse and fine labels. This framework can not only generate hierarchical labels for images, but also improve the traditional leaf-level classification performance due to incorporating the hierarchical information. Moreover, this framework can be built on top of any CNN architecture which is primarily designed for leaf-level classification. Accordingly, we build a high performance network based on the CNN-RNN paradigm which outperforms the original CNN (wider-ResNet) and also the current state-of-the-art. In addition, we investigate how to utilize the CNN-RNN framework

to improve the fine category classification when a fraction of the training data is only annotated with coarse labels. Experimental results demonstrate that CNN-RNN can use the coarse-labeled training data to improve the classification of fine categories, and in some cases it even surpasses the performance achieved by fully annotated training data. This reveals that, CNN-RNN can alleviate the challenge of specialized and expensive annotation of fine labels.

## 5.1 Introduction

Image classification has long been an active area of research, which aims to classify images into pre-defined categories, and helps people to know what kind of objects the images contain. Traditionally, image classification is mainly performed on small datasets, by encoding local hand-crafted features and using them as input for classifiers [31, 122, 135].

In recent years, two fundamental changes occurred for this task: first, the number of digital images has been increasing exponentially. This brings people more alternatives, and more difficulties, in finding relevant images from this large volume of data. To help people access data, in an effortless and meaningful way, we need a good semantic organization of the categories. Second, deep learning methods have proven to be successful for image classification. In recent years, researchers have built various deep structures [24, 25, 227], and have achieved quite accurate predictions on small datasets [236, 259].

As a consequence, the current research focus has moved to larger and more challenging datasets [260, 261], such as ImageNet [149]. Such datasets often organize the large number of categories in a hierarchy, according to their semantic belongings. The deeper one goes in the hierarchy, the more specific the category is. In contrast to current approaches, which only focus on the leaf categories, we argue that generating hierarchical labels in a coarse-to-fine pattern can present how the semantic categories evolve, and thus can better describe what the objects are. For example, for Figure 5.1(c), the predicted leaf-category label is 'Triceratops'. Without specialized knowledge, we cannot learn that this category label belongs to the higher level category label 'Horned Dinosaur'.

The **first contribution** of this work is a framework capable of generating hierarchical labels, by integrating the powerful Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). CNN is used to generate discriminative features, and RNN is used to generate sequential labels.

There are several notable advantages for the CNN-RNN framework:

(a) Dog → Basenji (b) Dog → Leonberg (c) Horned Dinosaur → Triceratops

**Figure 5.1:** The example images with the 'coarse → fine' label.

(1) Learning things in a hierarchical way is consistent with human perception and concept organization. By predicting the labels in a coarse-to-fine pattern, we can better understand what the objects are, such as depicted in Figure 5.1(c).

(2) It can exploit the relationship between the hierarchical categories, which, in turn, helps the traditional image classification task. For example, when we build the CNN-RNN framework with wrn-28-10 [262], we can increase the accuracy of coarse and fine categories by 2.8% and 1.68%, respectively. To the best of our knowledge, this is the first work trying to employ RNN to improve the classification performance by exploiting the relationship between hierarchical labels.

(3) It is transferrable. In principle, the framework can be built on top of any CNN architecture which is primarily intended for single-level classification, and boost the performance for each hierarchical level. To verify this, we have conducted extensive experiments with three high-performance networks, i.e. CNN-7 [263], wrn-28-10 [262] and our proposed wider-Resnet.

(4) The structure can be trained end-to-end. In contrast to other methods which can only model the category relationship with pre-computed image features [264, 265], the CNN-RNN framework can jointly learn the features and relationship in an end-to-end way, which can improve the final predictions considerably. For a subset of ImageNet 2010, we compared employing pre-computed CNN features to train the RNN with end-to-end training the CNN and RNN and demonstrated a significant improvement of the subcategory accuracy from 77.27% to 82%.

(5) The number of the hierarchical labels can be variable. The flexibility of RNN allows us to generate hierarchical labels of different lengths, i.e. more

specific categories would have more hierarchical labels. We have demonstrated this property on the widely used ImageNet 2012 dataset [21].

As the framework is transferrable, we intend to build a high performance CNN model, and utilize its CNN-RNN variant to further boost the accuracy.

Recently, deep residual networks (ResNet) [227] have attracted great attention because of its leading performance in several image classification tasks and Zagoruyko et al. [262] further presented a thorough experimental study about several important aspects of ResNet, such as the width and depth, and proposed a wide Resnet that obtained better performance than the original ResNet. The **second contribution** of this work is, we build a wider ResNet compared to [262]. Our implementation shows that, the wider-Resnet performs better than [262] on CIFAR-100, and also outperforms the original ResNet with thousands layers. In addition, by utilizing the CNN-RNN framework, we obtain considerably better results than the state-of-the-art.

The performance of deep models has benefited from the accurate and large-scale annotations, such as ImageNet [149]. However, manual labeling is an excessively tedious and expensive task, especially for the fine-grained classes, which often require expert knowledge (e.g. breeds of dogs, flower species, etc.). For example, for Figure 5.1 (a) and Figure 5.1 (b), it is easy to annotate the images with the coarse label 'dog', but it requires specialized knowledge to divide them into subcategories 'Bsenji' and 'Leonberg'. One optional thought is, if a part of the training data is only annotated with coarse category labels, whether we could utilize the coarse-labeled training data to improve the classification performance of fine categories?

The **third contribution** of this work is, we investigate how to utilize the CNN-RNN framework to improve the subcategory classification when a fraction of the training data only has coarse labels. By training the CNN-RNN framework on the fully annotated data in the training set, we can exploit the relationship between the coarse and fine categories. Thereby, we can predict the fine labels of the coarse-labeled training data, and then re-train the CNN-RNN model. Experimental results demonstrate that the coarse-labeled training data can normally

help the subcategory classification. In some cases, it can even surpass the performance of fully annotated training data. This alleviates the expensive process of fine-grained labeling.

## 5.2 Related Work

### 5.2.1 Usage of CNN-RNN framework

Deep learning methods have attracted significant attention [226] and achieved revolutionary successes in various applications [213, 227]. Two important structures for deep learning are CNN and RNN. CNN has proven to be successful in processing image-like data, while RNN is more appropriate in modeling sequential data. Recently, several works [266–271] have attempted to combine them together, and built various CNN-RNN frameworks. Generally, the combination can be divided in two types: the unified combination and the cascaded combination.

The unified combination often attempts to introduce a recurrent property into the traditional CNN structure in order to increase the classification performance. For example, Zuo et al. [269] converted each image into 1D spatial sequences by concatenating the CNN features of different regions, and utilized RNN to learn the spatial dependencies of image regions. Similar work appeared in [272]. The proposed ReNet replaced the ubiquitous convolutional+pooling layer with four recurrent neural networks that sweep horizontally and vertically in both directions across the image. In order to improve the multi-label classification, Wang et al. [271] presented the CNN-RNN framework to learn a joint embedding space in modeling semantic label dependency as well as the image-label relevance.

On the other hand, the cascaded combination would process the CNN and RNN separately, where the RNN takes the output of CNN as input, and returns sequential predictions of different timesteps. The cascaded CNN-RNN frameworks are often intended for different tasks, rather than image classification. For example, [10, 266, 268] employed CNN-RNN to address the image captioning task, and [273] utilized CNN-RNN to rank the tag list based on the visual importance.

In this chapter, we propose to utilize the cascaded CNN-RNN framework to address a new task, i.e. hierarchical image classification, where we utilize CNN to generate discriminative image features, and utilize RNN to model the sequential relationship of hierarchical labels.

### 5.2.2 Hierarchical models for image classification

Hierarchical models have been used extensively for image classification. For example, Salakhutdinov et al. [274] presented a hierarchical classification model to share features between categories, and boosted the classification performance for objects with few training examples. Yan et al. [215] presented a hierarchical deep CNN that consists of a coarse component trained over all classes as well as several fine components trained over subsets of classes. Instead of utilizing a fixed architecture for classification, Murdock et al. [275] proposed a regularization method, i.e. Blockout, to automatically learn the hierarchical structure.

Another pipeline to employ hierarchical models tends to improve the classification performance by exploiting the relationship of the categories in the hierarchy. For instance, Deng et al. [264] introduced HEX graphs to capture the hierarchical and exclusive relationship between categories. Ristin et al. [265] utilized Random Forests and proposed a regularized objective function to model the relationship between the categories and subcategories. This type of hierarchical models can not only improve the traditional image classification performance, but also provide an alternative way to utilize the coarse-labeled training data.

In contrast to previous works, our work utilizes RNN to exploit the hierarchical relationship between coarse and fine categories, and aims to adapt the model to address the hierarchical image classification task, in which we simultaneously generate hierarchical labels for the images. Compared with [264, 265] that can only process the pre-computed image features, our proposed CNN-RNN framework can be trained end-to-end.

## 5.3 Hierarchical Image Classification

The goal of our approach is to simultaneously generate hierarchical labels of the images. To this end, we can employ two types of generators: a CNN-based generator and a CNN-RNN generator. Both of them keep the preceding layers of the basic CNN structure except for the last layer.

### 5.3.1 CNN-based generator

A CNN-based generator aims to generate coarse and fine labels by utilizing the conventional CNN structure. It acts as the general practice to fulfill this specific task. We replace the last layer of conventional CNN with two layers, through which to provide separate supervisory signals for both the coarse categories and fine categories. The two layers can be arranged either in a serial pattern (Figure 5.2: Strategy 1 & 2), or in a parallel pattern (Figure 5.2: Strategy 3).

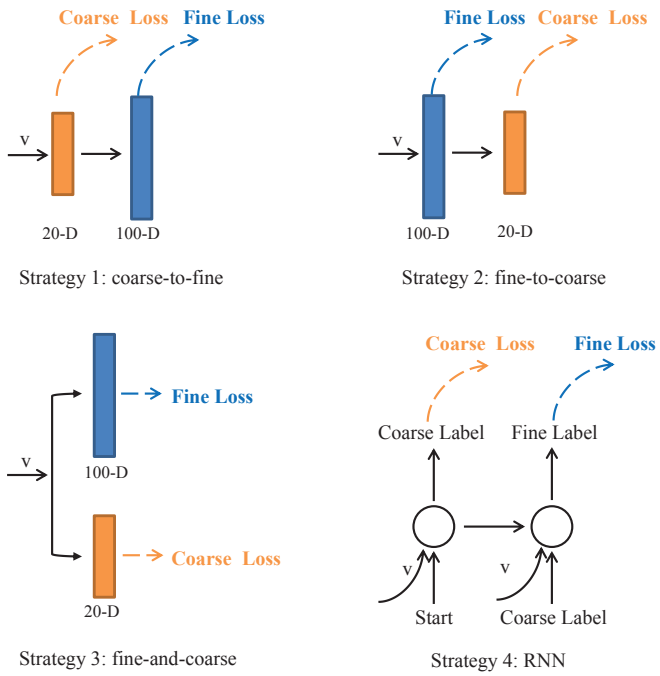



**Figure 5.2:** The illustration of the four strategies which can jointly train and generate the coarse and fine labels.

During the training phase, we utilize the softmax loss function to jointly optimize the coarse and fine label predictions, as defined in Eq 5.1.

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{j=1}^{C} 1\left\{x^i = j\right\}\log p_j + \sum_{k=1}^{F} 1\left\{y^i = k\right\}\log p_k\right) \tag{5.1}$$

Where $1\{\cdot\}$ is the indicator function. $N, C, F$ denote the number of the images, coarse categories, and fine categories, respectively. $p_j$ and $p_k$ are the softmax probabilities of the coarse and fine categories, respectively.

During the inference phase, we can utilize the trained network to determine the coarse and fine labels at the same time.

There are two potential drawbacks for the CNN-based generator: first, it treats the two supervisory signals individually, and does not exploit the relationship between them. Second, when the hierarchy is of variable length, we cannot define a universal CNN-based generator to simultaneously determine the hierarchical labels.

### 5.3.2 CNN-RNN generator

A CNN-RNN generator determines hierarchical predictions using an architecture where the last layer of CNN is replaced by RNN (Figure 5.2: Strategy 4).

RNN [276] is a class of artificial neural networks where connections between units form a directed cycle, as shown in Figure 5.3. It can effectively model the dynamic temporal behavior of sequences with arbitrary lengths. Long-Short Term Memory (LSTM) [277] is a particular form of a traditional RNN. It extends the cell in a standard RNN by using three gates, including an input gate, a forget gate and an output gate, to accumulate or forget relevant contextual information in its hidden state, as shown in Figure 5.3. These gates enable LSTM to model long-term dependencies in a sequence, and effectively address the gradient vanishing/exploding issues that commonly appear during RNN training [271].

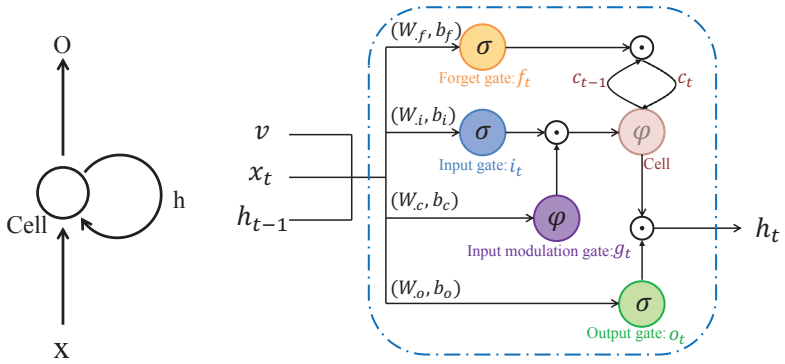



**Figure 5.3:** The pipeline of RNN(left) and LSTM(right).

In this work, we use LSTM neurons as our recurrent neurons. The definition of the gates and the update of LSTM at the timestep t are as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{vi}v + b_i) \tag{5.2}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{vf}v + b_f) \tag{5.3}$$

$$o_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{vo}v + b_o) \tag{5.4}$$

$$g_t = \varphi(W_{xc}x_t + W_{hc}h_{t-1} + W_{vc}v + b_c) \tag{5.5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{5.6}$$

$$h_t = o_t \odot \varphi(c_t) \tag{5.7}$$

Where $\odot$ represents the product operation, $\sigma$ is the sigmoid function ($\sigma(x) = (1+\exp(-x))^{-1}$), and $\varphi$ is the hyperbolic tangent function ($\varphi(x) = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$). The definition for other symbols are: $i_t$, $f_t$, $o_t$, $g_t$ denote the input gate, forget gate, output gate, and input modulation gate, respectively. $x$, $h$, $v$ and $c$ represent the input vector, hidden state, image visual feature, and memory cell, respectively. We propose to impose the image visual feature $v$ at each timestep when updating the LSTM. $W$ and $b$ are the weights and bias that need to be learned.

The goal of our approach is to generate hierarchical labels for images. The labels are ordered in a coarse-to-fine pattern, i.e. coarser labels appear at the front of the list. To this end, we merge the $C$ coarse categories and $F$ fine categories as $C + F$ super categories. For different timesteps, the CNN-RNN generator takes the labels of different levels as input, where the coarser-level labels appear

at the preceding timesteps. In this way, the coarser-level labels can provide insightful information for the prediction of finer labels. The procedure is shown in Figure 5.4.

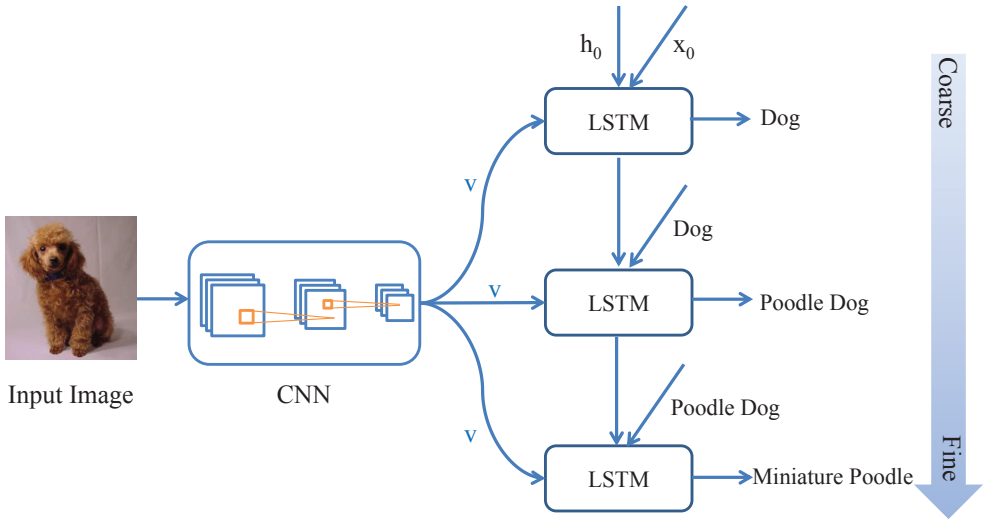



**Figure 5.4:** The pipeline of the CNN-RNN framework.

During the training phase, the CNN-RNN generator utilizes the groundtruth coarser-level labels as input, and jointly optimizes the coarse and fine predictions, as denoted in Eq 5.8.

$$\text{Loss} = -\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\sum_{j=1}^{C+F} 1\left\{x_t^i = j\right\}\log p_j\right) \tag{5.8}$$

During the inference phase, when the groundtruth coarser-level labels are not available, the CNN-RNN generator first predicts the maximum likelihood label for current timestep, i.e. $W_t = \text{argmax}_{W_{t-1}}\, p(W_{t-1}|I)$, and then utilize the predicted label as the input for the next timestep.

As the CNN-RNN generator defines the super categories, and it equally trains and predicts the super categories, we do not need to design specific networks for the categories of different levels. Therefore, the CNN-RNN generator is robust, and can be employed to generate hierarchical labels of different lengths.

## 5.4 Experiments

We perform our experiments on three well-known datasets: CIFAR-100 [278], ImageNet 2012 [21] and a subset of ImageNet 2010 [265]. These three datasets have provided hierarchical image labels. The characteristics of the three datasets are summarized in Table 5.1.

**Table 5.1:** The characteristics of the datasets, including the depth of the hierarchy, the number of the coarse categories and fine categories.

| Dataset | Depth | Coarse No. | Fine No. |
|---|---|---|---|
| CIFAR-100 | 2 | 20 | 100 |
| ImageNet 2012 | 1-9 | 860 | 1000 |
| Subset of ImageNet 2010 | 2 | 143 | 387 |

The performance is measured based on the top-1 accuracy. All the experiments are conducted using the Caffe [218] library with a NVIDIA TITAN X card.

The experiments can be divided into two parts. In the first part, we evaluate the performance of hierarchical predictions. In the second part, we investigate the performance of subcategory classification when only a part of the training data is labeled with fine labels while the rest only has coarse labels.

### 5.4.1 Hierarchical predictions

We evaluate the hierarchical predictions on two widely-used datasets: CIFAR-100 [278] and ImageNet 2012 [21].

#### 5.4.1.1 CIFAR-100

CIFAR-100 contains 100 classes, and each class has 500 training images and 100 test images. These classes are further grouped into 20 superclasses. Therefore, each image comes with two level labels: a fine label (the class to which it belongs) and a coarse label (the superclass to which it belongs). For data preprocessing,

we normalize the data using the channel means and standard deviations. The symbol '+' means a standard data augmentation, i.e. first zero-padded with 4 pixels on each side, and randomly crop $32 \times 32$ images from the padded images, or their horizontal reflections.

**Evaluation of the hierarchical image classification task.**
The CNN-based generator and CNN-RNN generator are considered as two alternative structures to fulfill the hierarchical image classification task. In contrast to CNN-based generator, the CNN-RNN generator can effectively exploit the dependency of the hierarchical labels, and thereby achieving a better classification performance for both the coarse and fine categories. We compare their performance in Table 5.2.

**Table 5.2:** The comparison of the accuracy for the coarse categories and fine categories. '+' indicates a standard data augmentation (translation/mirroring)

| | C100 | | C100+ | |
|---|---|---|---|---|
| | coarse | fine | coarse | fine |
| coarse-to-fine | 73.88% | 58.41% | 78.1% | 64.16% |
| fine-to-coarse | 75.02% | 61.75% | 78.16% | 65.56% |
| fine-and-coarse | 74.72% | 61.8% | 77.56% | 64.87% |
| CNN-RNN | **80.81%** | **69.69%** | **83.21%** | **72.26%** |

All of the evaluations in this part are conducted based on the CNN model proposed in [263], because of its high training efficiency and decent performance on CIFAR-100. The CNN structure is shown in Figure 5.5. We employ exactly the same experimental configuration as used in [263].

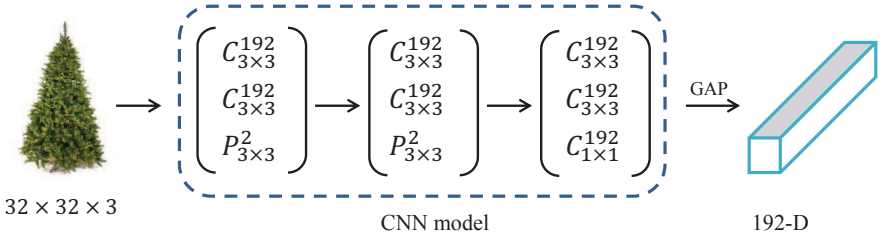



**Figure 5.5:** The CNN baseline proposed in [263].

As can be seen, the CNN-RNN generator can significantly outperform the CNN-based generator, both for the coarse/fine predictions with/without data augmentation. Specifically, for the coarse predictions, the CNN-RNN generator outperforms the CNN-based generator by at least 5.05%, while for the fine predictions, the CNN-RNN generator is even more advantageous, with an improvement of more than 6.7%. This demonstrates that, by exploiting the latent relationship between the coarse and fine categories, RNN can properly address the hierarchical-based task.

**Evaluation of the traditional image classification task.**
The traditional image classification task consists of classifying images into one pre-defined category, rather than multiple hierarchical categories.

As the CNN-RNN generator can simultaneously generate the coarse and fine labels, in this part, we further compare its performance with 'coarse-specific' and 'fine-specific' networks. The 'fine-specific' network uses the common CNN structure which is specifically employed for the fine category classification. The 'coarse-specific' network shares the same preceding layers with the 'fine-specific' network, where the last layer is adapted to equal the coarse category number, e.g. 20 for CIFAR-100.

The coarse-specific, fine-specific and CNN-RNN framework can be constructed based on any CNN architecture. To make the comparison more general and convincing, we evaluate the performance on three networks: CNN-7 [263], wrn-28-10 [262] and our proposed wider-Resnet.

For wrn-28-10, we adopt the version with dropout [52], and train the network with larger mini-batch size (i.e. 200), and more iterations (a total of $7 \times 10^4$ iterations, and the learning rate dropped at $2 \times 10^4$, $4 \times 10^4$, $6 \times 10^4$ iterations). Other experimental configuration follows [262].

The structure of our proposed wider-Resnet is shown in Table 5.3. We adopt the pre-activation residual block as in [279], and train the models for a total of $7 \times 10^4$ iterations, with a mini-batch size of 200, a weight decay of 0.0005 and a

momentum of 0.9. The learning rate is initialized with 0.1, and is dropped by 0.1 at $4 \times 10^4$ and $6 \times 10^4$ iterations.

**Table 5.3:** The framework of our proposed wider-Resnet.

| Group name | Output size | wider-Resnet |
|---|---|---|
| conv1 | 32×32 | 3×3, 64 |
| conv2 | 32×32 | $\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 256 \end{bmatrix} \times3$ |
| conv3 | 16×16 | $\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 512 \end{bmatrix} \times3$ |
| conv4 | 8×8 | $\begin{bmatrix} 1\times1, 1024 \\ 3\times3, 1024 \\ 1\times1, 1024 \end{bmatrix} \times3$ |
| pool5 | 1×1 | global average pooling |

The results on these three datasets are shown in Table 5.4.

**Table 5.4:** The comparison of the accuracy for the coarse categories and fine categories. '+' indicates standard data augmentation (translation/mirroring)

| | | C100+ | |
|---|---|---|---|
| | | coarse | fine |
| CNN-7 [263] | coarse-specific | 82.09% | - |
| | fine-specific | - | 72.03% |
| | CNN-RNN | **83.21%** | **72.26%** |
| wrn-28-10 [262] | coarse-specific | 82.59% | - |
| | fine-specific | - | 74.55% |
| | CNN-RNN | **85.39%** | **76.23%** |
| wider-Resnet | coarse-specific | 85.38% | - |
| | fine-specific | - | 77.97% |
| | CNN-RNN | **88.23%** | **79.14%** |

We can see that, CNN-RNN can simultaneously generate the coarse and fine labels without developing two separate models, and the accuracy for both categories outperforms the specific networks. Take our proposed wider-Resnet as an example, the CNN-RNN structure increases the coarse and fine accuracy by 2.85% and 1.17% respectively, over the coarse-specific and fine-specific networks. This advantage demonstrates that, by exploiting the latent relationship of the coarse and fine categories, CNN-RNN can help the traditional image classification task.

Our implementation of wrn-28-10 [262] cannot reproduce the original published results, possibly as a result of the differences in the platforms (Torch v.s. Caffe), or the differences in the preprocessing step (pad with reflections of original image v.s. pad with zero). Nevertheless, we can still improve the coarse and fine accuracy by 2.8% and 1.68% respectively, through utilizing the CNN-RNN structure.

**Comparison with the state-of-the-art.**
We compare our wider-Resnet network, as well as its CNN-RNN variant, with the state-of-the-art, as is shown in Table 5.5.

Through the comparison, we further demonstrate the superiority of the wider networks on CIFAR-100 dataset, as our not-very-deep wider-Resnet network (29 layers) surpasses the performance of the ResNet with super deep layers (1001 layers). In comparison with another wide ResNet [262] with similar depth, wider-Resnet also demonstrates great improvements and remarkably reduces the classification error from 25.45% to 22.03%, under the same platform and pre-processing step.

Overall, our proposed wider-Resnet achieves the best performance over previous works, and wider-Resnet-RNN further increases the state-of-the-art to 20.86%. Nevertheless, we are still seeking to build the CNN-RNN framework on top of future state-of-the-art architectures to boost the classification performance.

**Table 5.5:** The test error of different methods on the CIFAR-100 dataset with standard data augmentation (translation/mirroring)

| Method | C100+ |
|---|---|
| FitNet [280] | 35.04% |
| DSN [281] | 34.57% |
| All-CNN [282] | 33.71% |
| Highway Network [283] | 32.39% |
| APL [284] | 30.83% |
| SReLU [285] | 29.91% |
| BayesNet [286] | 27.4% |
| Fitnet4-LSUV [287] | 27.66% |
| ELU [288] | 24.28% |
| MBA [289] | 24.1% |
| ResNet-110 [227] (according to [290]) | 27.22% |
| ResNet-110 (Stochastic Depth) [290] | 24.58% |
| ResNet-164 (Pre-activation) [279] | 24.33% |
| ResNet-1001 (Pre-activation) [279] | 22.71% |
| 18-layer + wide RiR [291] | 22.90% |
| FractalNet-20 [292] | 23.30% |
| FractalNet-40 [292] | 22.49% |
| SwapOut V2 [293] (width×4) | 22.72% |
| wrn-28-10 [262](our reproduced) | 25.45% |
| wrn-28-10-RNN | 23.77% |
| wider-Resnet | 22.03% |
| wider-Resnet-RNN | **20.86%** |

#### 5.4.1.2 ImageNet 2012

One notable advantage of RNN is that it can generate sequences with variable lengths. To demonstrate this, we investigate the CNN-RNN framework on the widely used ImageNet 2012 dataset [21].

ImageNet is an image dataset organized according to the WordNet hierarchy [294]. It is larger in scale and diversity than other image classification datasets. ImageNet 2012 uses a subset of ImageNet with roughly 1300 images in each of 1000

categories. The images are annotated with hierarchical labels of different lengths. In total, there are about 1.2 million training images and 50000 validation images. For all the experiments, we train our model on the training images, and test on the validation images of the ImageNet 2012 dataset.

We utilize the ResNet-152 [227] as our CNN model. For simplicity, pre-trained model weights are kept fixed without fine-tuning. For the RNN model, we use 1000 dimensions for the embedding and the size of the LSTM memory. During the experiments, we first resize all the images to $224 \times 224$ pixels and extract the last pooling features utilizing ResNet-152, and then send the features into LSTM for modeling the category dependency.

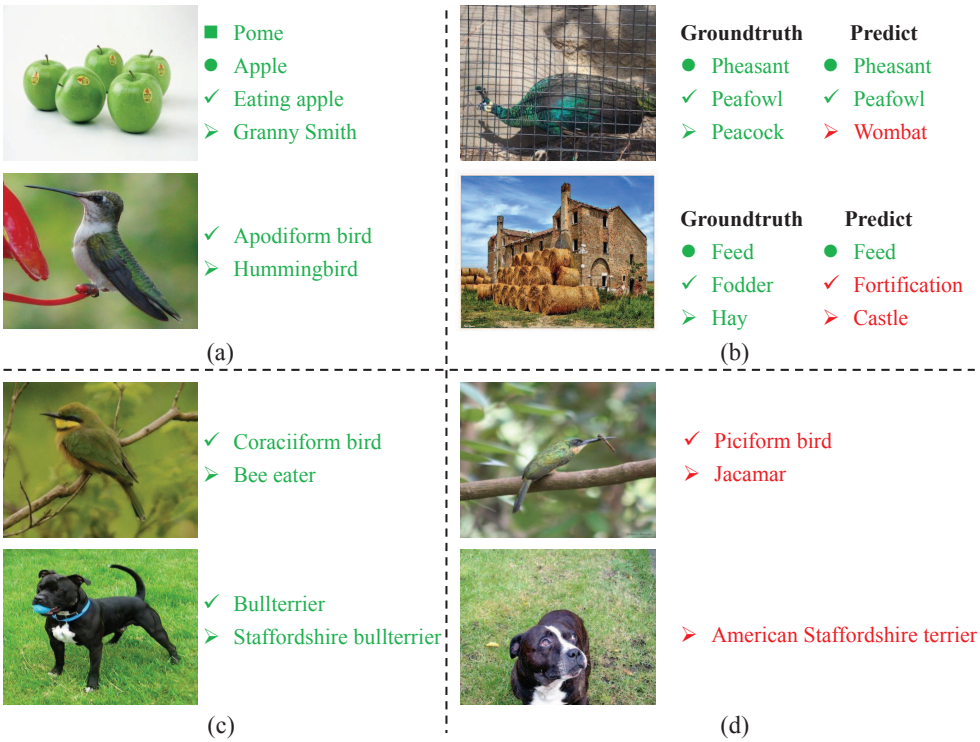



**Figure 5.6:** The hierarchical predictions of some example images. (a) and (c) show some positive examples. (b) shows the examples with partly wrong predictions, e.g. correct coarse labels & wrong fine labels. (d) shows examples in the same category as (c), but which have totally wrong predictions

Figure 5.6 demonstrates the hierarchical predictions for some example images,

from which we can observe that: First, RNN is able to generate predictions with different lengths, and more specific categories would have more hierarchical labels. Second, the hierarchical labels can describe how the fine categories are evolved from higher level coarse categories, and thus can provide us a better understanding of the objects. Consider for example the upper image in Figure 5.6 (a), we may get confused with the leaf-level label: 'Granny Smith'. But when the coarse-level labels are provided, we observe that 'Granny Smith' is a breed of apple. Third, it may be more difficult to classify images into leaf-level categories than branch-level categories. When we get faulty leaf-level predictions for the given image, we might still learn what the image depicts from the coarse predictions, as shown in Figure 5.6 (b).

### 5.4.2 From coarse categories to fine categories

In the previous section, we have investigated the hierarchical classification performance of CNN-RNN when all of the coarse and fine labels are available for the training data. However, annotating fine labels for large amounts of training data is quite expensive, especially when it requires expert knowledge. In this subsection, we focus on a scenario in which a part of the training data is annotated with fine labels, while the rest only has coarse labels. This can be viewed as a special case of weakly supervised learning, and has ever been investigated in [265].

We follow the experiment setup of [265], and conduct our experiment on a subset of ImageNet 2010. This dataset particularly selected the classes from ImageNet 2010 that have a unique parent class, and obtained 143 coarse classes and 387 fine ones accordingly. The reduced training set contains 487K images where each coarse class has between 1.4K and 9.8K images, and each fine class has between 668 and 2.4K images. The test set contains 21450 images, and each coarse class has 150 images [1].

All of the image features are extracted from the VGG-Net [24], as was done for the preliminary experiments in [265].

[1] More details about the dataset are available at
$http://www.vision.ee.ethz.ch/datasets_extra/mristin/ristin_etal_cvpr15_data.zip$

**Evaluation of the classification performance when all of the training fine labels are available**

When all of the coarse and fine labels are available, we can directly train the RNN on the full training set, and evaluate the classification performance on the test set. To better demonstrate the advantage of RNN, we further conduct the training process on a fraction of the training set. In addition, we investigate how much the performance may improve when the coarse labels are provided for the test data, and when we train the CNN-RNN in an end-to-end way, rather than with the off-the-shelf image features. As a comparison with CNN-RNN framework, we also finetune the VGG-Net on the ImageNet 2010 subset. The results are shown in Table 5.6.

**Table 5.6:** Accuracy for classifying fine labels using the ImageNet 2010 subset described in [265]. RNN: train the RNN with extracted image features from VGG-Net [24]; CNN: finetune the VGG-Net [24] on the ImageNet 2010 subset; CNN-RNN: jointly train the VGG-Net[24] and RNN in an end-to-end pattern; We use the superscript '$*$' to denote that the coarse labels are provided when predicting the fine labels in the test phase.

| | Training Set | Accuracy |
|---|---|---|
| NCM [295] | $S$ | 66.02% |
| Multiclass SVM [296] | $S$ | 71.67% |
| RNCMF [265] | $S$ | 74.18% |
| RNN | $0.2S$ | 75.09% |
| | $0.4S$ | 76.17% |
| | $S$ | 77.27% |
| CNN | $S$ | 76.01% |
| CNN-RNN | $S$ | 82% |
| CNN-RNN$^*$ | $S$ | 90.69% |

We can notice that, training on more data results in a more powerful RNN model, and thus can achieve better performance. Compared with the models trained on parts of the training set, i.e. $0.2S$ and $0.4S$, utilizing the full training set $S$ shows an improvement of 2.18% and 1.1%, respectively. It reveals that, a large training dataset is essential in training the deep models.

In contrast to other methods listed in [265], RNN achieves superior classification performance by inherently exploiting the relationship between the coarse and fine categories. Notably, RNN can deliver better performance even utilizing only 20 percent of the training data.

One additional advantage of the CNN-RNN framework is that it can be trained end-to-end. Compared with the predictions generated with off-the-shelf CNN features, jointly training the CNN and RNN results in a significant improvement, from 77.27% to 82%. It is also much better than directly finetuning the VGG-Net on the ImageNet 2010 subset (82% v.s. 76.01%). When provided the coarse labels for the test images, CNN-RNN achieves an accuracy of 90.69%.

**Evaluation of the classification performance when part of the fine labels for training are missing**

The training set $S$ in this part are randomly divided into two disjoint sets: $S_{\text{coarse}}$ and $S_{\text{fine}}$. $S_{\text{coarse}}$ has only the coarse labels, while $S_{\text{fine}}$ has both coarse and fine labels. We vary $|S_{\text{fine}}| \in \{0.1|S|, 0.2|S|, 0.5|S|\}$, and for each $S_{\text{fine}}$, we further vary $|S_{\text{coarse}}| \in \{0.1|S|, 0.2|S|, 0.5|S|\}$.

For each training/test configuration, we conduct three evaluations:

1) $S_{\text{fine}}$: We train the RNN on $S_{\text{fine}}$, and evaluate on the test set;

2) $S_{\text{fine}} + S^{-}_{\text{coarse}}$: We first train the RNN on $S_{\text{fine}}$, and use it to predict the fine labels of $S_{\text{coarse}}$. In this way, we obtain a new training set $S^{-}_{\text{coarse}}$, which contains both coarse and (predicted) fine labels. Next, we utilize the $S_{\text{fine}}$ and $S^{-}_{\text{coarse}}$ to re-train the RNN, and evaluate on the test set.

3) $S_{\text{fine}} + S^{+}_{\text{coarse}}$: We train the RNN on $S_{\text{fine}}$ and $S^{+}_{\text{coarse}}$, and evaluate on the test set. $S^{+}_{\text{coarse}}$ means we utilize the groundtruth fine labels of $S_{\text{coarse}}$.

The results are shown in Figure 5.7.

In general, $S_{\text{fine}} + S^{-}_{\text{coarse}}$ performs better than $S_{\text{fine}}$, indicating that even some of the fine labels for the training data are missing, the fine category classification can benefit from the CNN-RNN structure.

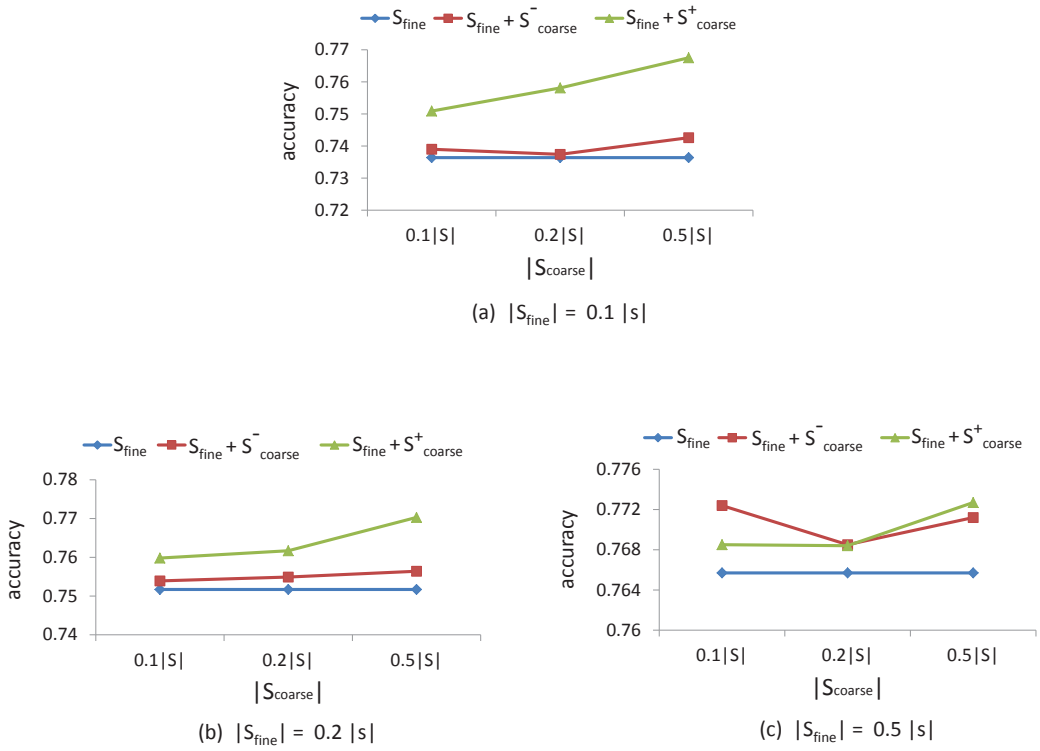



**Figure 5.7:** The classification performance with different training/test set.

Since the fine labels of $S_{\text{coarse}}$ are predicted by the RNN trained on $S_{\text{fine}}$, their accuracy cannot be guaranteed. As a consequence, the second training of RNN may be conducted on a partly wrong labeled dataset. This is particularly severe when $|S_{\text{fine}}|$ is small. As we can see in Figure 5.7(a), when $|S_{\text{fine}}| = 0.1|S|$, the classification hardly benefited from using $S_{\text{coarse}}$ when compared to the RNN trained solely on $S_{\text{fine}}$.

On the contrary, when $|S_{\text{fine}}|$ is large, e.g. $|S_{\text{fine}}| = 0.5|S|$, we can achieve a considerable improvement by incorporating $S_{\text{coarse}}$. Notably, when $|S_{\text{fine}}| = 0.5|S|, |S_{\text{coarse}}| = 0.1|S|$, $S_{\text{fine}} + S^{-}_{\text{coarse}}$ even performs slightly better than $S_{\text{fine}} + S^{+}_{\text{coarse}}$, demonstrating its great potential in weakly supervised classification.

We further compare our method with the NN-H-RNCMF [265], which also attempted to improve the classification by exploiting the hierarchy. We set the amount of coarse-labeled data to $|S_{\text{coarse}}| = 0.5|S|$, and vary the amount of fine-labeled data $|S_{\text{fine}}| \in \{0.1|S|, 0.2|S|, 0.5|S|\}$, and the results are shown in Ta-

ble 5.7. It can be seen that, RNN performs much better than NN-H-RNCMF in all configurations, demonstrating its great potential in exploiting the hierarchical relationship.

**Table 5.7:** Accuracy in classifying fine categories for the test set. We set the amount of coarse-labeled data to $|S_{\text{coarse}}| = 0.5|S|$.

| | $\|S_{\text{fine}}\|$ | | |
|---|---|---|---|
| | $0.1\|S\|$ | $0.2\|S\|$ | $0.5\|S\|$ |
| RNCMF [265] | 68.49% | 70.49% | 73.07% |
| NN-H-RNCMF [265] | 69.95% | 71.41% | 73.43% |
| RNN | **74.26%** | **75.64%** | **77.12%** |

## 5.5 Conclusion

In this chapter, we proposed to integrate CNN and RNN to accomplish hierarchical classification task. The CNN-RNN framework can be trained end-to-end, and can be built on top of any CNN structures that are primarily intended for leaf-level classification, and further boost the prediction of the fine categories. In addition, we also investigated how the classification would benefit from coarse-labeled training data, which alleviates the professional and expensive manual process of fine-grained annotation.

Currently, it is necessary to have hierarchical labels in the training set, in order to train the RNN. However, this is not available for many small datasets. In the future, we will examine taking advantage of traditional clustering methods towards automatically constructing a hierarchy for the objects, and use CNN-RNN to boost the classification performance for general datasets.

# Chapter 6

# What Convnets Make for Image Captioning?

Nowadays, a general pipeline for the image captioning task takes advantage of image representation based on convolutional neural networks (CNNs) and sequence modeling based on recurrent neural networks (RNNs). Captioning performance closely depends on the discriminative capacity of CNNs. Our work aims to investigate the effects of different Convnets (CNN models) on image captioning. We train three Convnets based on different classification tasks: single-label, multi-label and multi-attribute, and feed the image features from these Convnets into a Long Short-Term Memory (LSTM) to model the sequence of words. Since the three Convnets focus on different visual contents in one image, we propose aggregating them together to generate a richer visual representation. Furthermore, during testing, we use an efficient multi-scale augmentation approach based on fully convolutional networks (FCNs). Extensive experiments on MS COCO 2014 dataset provide significant insights into the effects of Convnets. Moreover, we achieve comparable results to the state-of-the-art for both caption generation and image-sentence retrieval tasks.

## 6.1 Introduction

Image captioning is a fundamental and important task in vision-to-language research. It aims to describe an image with meaningful and sensible sentence-level captions. The automatically generated descriptions should cover the salient content in images, including objects, actions and other relations. In early research of image captioning, it has been converted to a retrieval-based task. Those retrieval-based approaches [297–299] focus on mapping images to sentences based on predefined captions. However, they fail to generate novel sentences for unseen scenes. To address this issue, generative approaches are developed to estimate novel sentences, such as Midge [300] and Baby Talk [301].

Recently, a new paradigm for image captioning is proposed in many state-of-the-art approaches [266, 267, 302–304]. This paradigm mainly integrates a convolutional neural network (CNN) and a recurrent neural network (RNN) together. The CNN is used to capture high-level image features, while the RNN generates a sequence of words based on the image features. In particular, a rich visual representation contributes much to generating accurate image captions. However, some Convnets (CNN models) are originally trained for image classification, but not for image captioning. It thus raises an important question: *What Convnets make for image captioning?*

Our aim in this work is to fully investigate the effects of different Convnets on image captioning. We exploit three kinds of Convnets: single-label Convenet, multi-label Convnet, and multi-attribute Convnet. (1) A single-label Convnet indicates a CNN model pre-trained on ImageNet dataset [21], such as AlexNet [14] and VGG-16 [24]. This Convnet can often offer one generic image representation. (2) A multi-label Convnet can predict multiple class labels given one image. It is consistent with the observation that sentence-level captions often talk about many salient objects jointly in images. Therefore, we fine-tune a multi-label Convnet on MS COCO 2014 [305] that consists of 80 object categories. Each image is annotated with multiple object labels. (3) A multi-attribute Convnet can not only reflect multiple object classes, but also describe actions and other relations about objects, for example jumping, sitting and interacting. Therefore,

a multi-attribute Convnet is able to narrow the gap between vision and language. We fine-tune a multi-attribute Convnet based on 300 attributes derived from MS COCO captions [305].

By observing the feature maps learned in the three Convnets, we find that their maps focus on different visual fields in images. Therefore, we propose aggregating their features together to generate a richer representation.

In addition, during the test stage, we take advantage of the efficient fully convolutional networks (FCNs) [60] for multi-scale augmentation. We use two scales of FCNs that are interpreted from one pre-trained CNN. This augmentation approach can be applied to both the single Convnet and multi-Convnet aggregation. Finally, we employ the Long Short-Term Memory (LSTM) [277] to build the language model. Figure 6.1 shows an image example from MS COCO 2014 [305]. Note that the visual feature is fed to the LSTM unit at each time step.

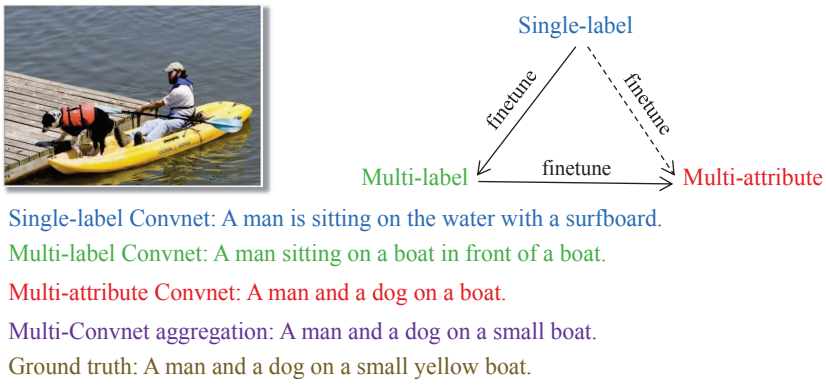



**Figure 6.1:** Example of image captioning using different Convnets. Each Convnet shows meaningful description. As compared to the human-written ground-truth, the multi-Convnet can generate closer result than any single Convnet.

In a nutshell, **our contributions** can be summarized as follows:

- We present a full comparison among the three Convnets for the image captioning task. Furthermore, we study the benefits of each Convnet and then integrate multiple Convnets for a richer visual representation. Our work can provide promising insights into deeply diagnosing and understanding Convnets for vision-to-language tasks.

- We employ an efficient multi-scale augmentation approach using FCNs.
- We achieve comparable results to the state-of-the-art on MS COCO 2014 dataset, both for caption generation and image-sentence retrieval tasks.

## 6.2 Related Work

In this section, we summarize related image captioning approaches based on CNN-RNN as below.

A prior work in NIC [267] employed a CNN-RNN scheme to model the image captioning problem. CNNs are used as the "encoder" to visually represent the input image with a fixed-length feature vector. Then RNNs, as the "decoder", can translate the feature vector into sentence-level captions. Similarly, other similar approaches [266, 304, 306] followed this CNN-RNN paradigm. Instead of only using CNN features, Jia et al. [307] added extra semantic information to each unit of the LSTM block. Jin et al. [308] integrated scene-specific contexts in order to highlight higher-level semantic information in images. In addition, Xu et al. [303] introduced a visual attention based model inspired by human visual system. The attention mechanism can automatically learn latent alignments between regions and words. Apart from the whole image captioning, there were some works focusing on image regions based captioning [302, 309, 310]. They first localized salient regions in images and then described them with natural language.

Recent work in [268] began capturing attributes to represent visual content. Notably, Yao et al. [311] investigated the performance upper bounds based on attributes for image and video captioning. However, both of these works did not train a new CNN model based on attributes. The most similar work in [312] fine-tuned a CNN based on the task of image-attribute classification. In comparison, our work had several main differences from [312]:

First, we intended to add a multi-label Convnet as a bridge from a single-label to a multi-attribute Convnet (see the two solid lines in Figure 6.1). Thus our multi-attribute Convnet had two-stage fine-tuning. In contrast, [312] directly fine-tuned a multi-attribute Convnet from a single-label Convnet (see the dash line

in Figure 6.1), and failed to study the effects of a multi-label Convnet. Second, we further evaluated the aggregation of multiple Convnets that has not been studied previously in [312]. Third, we presented an efficient multi-scale testing approach as compared to using expensive region proposals in [312]. In addition, their testing step was not end-to-end.

## 6.3 Proposed Approach

In this section, we will present our image captioning system in three aspects. First, we show the usage of single Convnet for capturing visual representation. Second, we find that integrating image features from the three Convnets is beneficial for a richer representation. Third, at the test stage, we use a multi-scale testing approach based on FCNs.

### 6.3.1 Convnets for Image Captioning

This part introduces the training details about the three Convnets. Notably, the multi-attribute Convnet also belongs to a multi-label classification task, but it has different training from the multi-label Convnet.

**Single-label Convnet.** CNNs trained on ImageNet dataset [21] are widely used as off-the-shelf feature extractors, such as Alexnet [14] and VGG-16 [24]. We call these CNNs as single-label Convnets, since they are originally trained for single-label classification, for example 1000 classes in ImageNet 2012. Here we use the VGG-16 net as a single-label Convnet for our image captioning system. As the left part in Figure 6.2, an image from MS COCO [305] is fed to a single-class Convnet that outputs a 1000-Dim visual feature.

**Multi-label Convnet.** Image captions often focus on multiple objects in images, instead of mentioning only one salient object. We thus train a multi-label Convnet on MS COCO 2014 dataset [305] that consists of 80 object categories. Each image in MS COCO is annotated by about 3 object labels on average. Instead of training from scratch, we fine-tune the single-label Convnet for a multi-label recognition

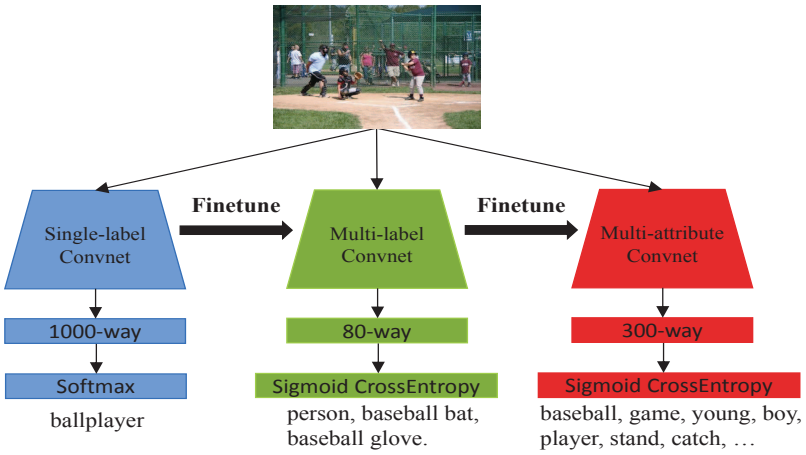



**Figure 6.2:** Illustration of the three Convnets for visual representations. The multi-label Convnet is fine-tuned from the pre-trained single-label Convnet. The multi-attribute Convnet performs two-stage fine-tuning.

task. Note that we replace the original 1000-way layer with 80-way layer. We use a sigmoid cross-entropy function to compute the element-wise loss. Assume that there are $K$ classes (e.g. 80), the total cost sums up $K$ of sigmoid losses by

$$l_1(x) = -\sum_{k=1}^{K} y_k(x) \log p_k(x) + (1 - y_k(x)) \log(1 - p_k(x)), \tag{6.1}$$

where $y_k \in \{0, 1\}$ is the ground-truth label indicating the absence or presence of the category $k$ in the input image $x$. $P_k(x)$ indicates the prediction probability of containing the category $k$. During fine-tuning, the parameters of the last fully-connected layer (i.e. the multi-class prediction layer) are initialized with gaussian filters. We initialize the learning rate of the last fully-connected layer with 0.01. Instead, the learning rates of other convolutional layers and fully-connected layers (i.e. fc6 and fc7) are initialized with 0.0001 and 0.001, respectively. The learning rate is divided by 10 after $2 \times 10^4$ iterations. The whole training will be terminated after $5 \times 10^4$ iterations. Besides, we use a weight decay of 0.0001, a momentum of 0.9, and a mini-batch size of 100. The multi-label Convnet is shown in the middle part in Figure 6.2.

**Multi-attribute Convnet.** Apart from object categories, a descriptive caption should mention more information like actions (e.g. sit, run) and other relations

(e.g. blue, small). Hence, using a Convnet that can reflect more attributes is beneficial for narrowing the gap between visual features and language words. Based on a multi-label Convnet, we further fine-tune a multi-attribute Convnet. First, we build an attribute dictionary based on MS COCO captions dataset. In [311], they summarize three groups of atoms: entity, action and attribute. We select top-100 atoms from each group, therefore, the attribute dictionary consists of 300 words (or attributes) in total. Note that the atoms defined in [311] are renamed as attributes in our work. Then, we remake the topmost layer with a 300-way fully-connected layer, as shown in the right part in Figure 6.2. Assume that $G$ denotes the number of attributes (e.g. $G = 300$). Similarly, the sigmoid cross-entropy loss is computed by

$$l_2(x) = -\sum_{g=1}^{G} y_g(x) \log p_g(x) + (1 - y_g(x)) \log(1 - p_g(x)), \tag{6.2}$$

where $y_g \in \{0, 1\}$ is the ground truth; $P_g(x)$ is the prediction probability. Since each image in MS COCO has five human-written captions, we merge five captions together to generate the ground-truth. During fine-tuning the multi-attribute model, we use the same hyper-parameters as the multi-label training.

To compare the visual features from the three Convnets, we visualize their most activated feature maps learned in the fifth convolutional layer (i.e. conv5_3), as illustrated in Figure 6.3. Here, we regard the feature map which has the largest average activation value as the most activated feature map. It can be seen that the three Convnets focus on different visual fields in images. This offers clear insights into diverse characteristics of the three Convnets.

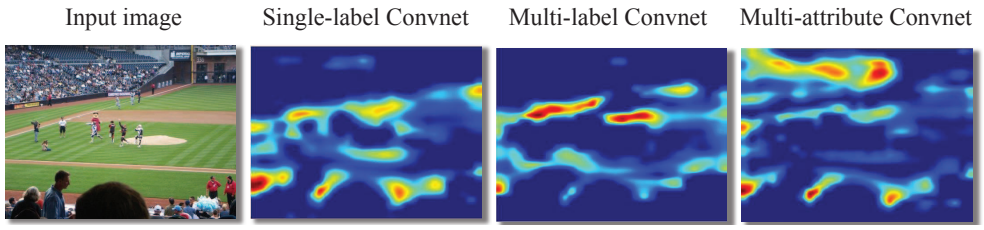



**Figure 6.3:** Visualization of feature maps for the three Convnets. We select the most activated feature map in conv5_3. We can see that the three Convnets focus on different visual fields in images due to their different classification objectives.

### 6.3.2 Multi-Convnet Aggregation

Since the three Convnets are trained for different classification objectives and can represent different features given the input image, we propose aggregating them together to compensate the deficiency of any single Convnet feature. Although a multi-attribute Convnet may contain the same objects as in a single-label and multi-label Convnet, the aggregation feature can further improve the accurate prediction of object classes. Figure 6.4 illustrates the pipeline of generating image captions based on multi-Convnet aggregation.

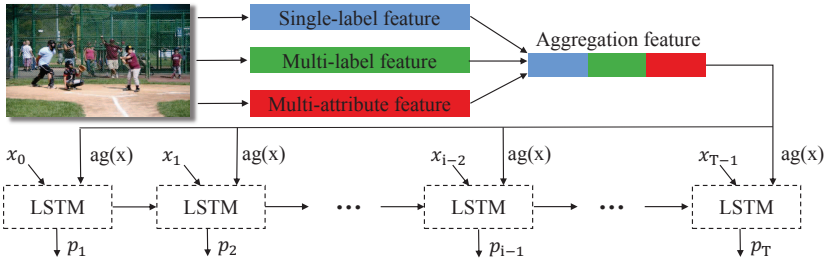



**Figure 6.4:** The pipeline of Image captioning based on multi-Convnet aggregation. The three Convnet features are concatenated together to generate an aggregation feature $ag(x)$. At each time step, both a word $x_i$ and $ag(x)$ are fed to the LSTM unit whose output is a probability distribution for the next word.

First, the input image $x$ is fed to three pre-trained Convnets to capture separate visual features, denoted as $sc(x), mc(x), ma(x)$. We then concatenate three kinds of features to create an aggregation feature $ag(x)$ (i.e. 1380-Dim vector), where $ag(x) = [sc(x), mc(x), ma(x)]$. Then, we add this aggregation feature to the following RNN unit at each time step. We employ one-layer Long Short-Term Memory (LSTM) [277] that can alleviate the vanishing gradient problem due to its gates mechanism. Finally, at the time step $t$, the formulation of LSTM units with an aggregation feature can be summarized as below

$$i_t = \sigma(W_{xi}x_t + W_{vi}ag(x) + W_{hi}h_{t-1} + b_i) \tag{6.3}$$

$$f_t = \sigma(W_{xf}x_t + W_{vf}ag(x) + W_{hf}h_{t-1} + b_f) \tag{6.4}$$

$$o_t = \sigma(W_{xo}x_t + W_{vo}ag(x) + W_{ho}h_{t-1} + b_o) \tag{6.5}$$

$$g_t = \phi(W_{xg}x_t + W_{vg}ag(x) + W_{hg}h_{t-1} + b_g) \tag{6.6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{6.7}$$

$$h_t = o_t \odot \phi(c_t) \tag{6.8}$$

$$p_{t+1} = Softmax(h_t) \tag{6.9}$$

where $W$and $b$ are the weight matrices and bias terms. We refer to $x_t$ as the input word at time step $t$ for image $x$. $\sigma$ and $\phi$ are the sigmoid and tangent activation functions. $p_{t+1}$ is used to predict the probability distribution for the next word. Finally, the objective in LSTMs for language modeling is to minimize the following loss cost

$$-\sum_{t=0}^{T-1} \log p_t(x_{t+1}|x_t, ag(x)) + \lambda||W||_2^2 \tag{6.10}$$

where $T$ is the length of the input sequence of words, and $\lambda$ indicates the weight decay (In this work, we follow the configuration of [266] and set $\lambda$ equals 0). For notational simplicity, we just give the computation of one input image and drop the mini-batch size in the formulation. Following the hyper-parameters in [266], both the word embedding size and hidden state size are set to 1000. We use a mini-batch size of 100 image-sentence pairs. The learning rate is initialized with 0.01 and decreases to one tenth of current rate after 20,000 iterations. The whole training will be terminated after 80,000 iterations. In addition, we use a momentum of 0.9 and clip gradients of 10.

### 6.3.3 Multi-scale Testing

During the test phase, we intend to use a multi-scale augmentation approach to capture a more robust image representation, as shown in Figure 6.5. We first extract a feature vector by inputting a 224×224 image to CNNs. Then, we convert one CNN model to a fully convolutional networks (FCN) [60]. FCN is quite efficient to compute regions based predictions without decreasing the ease of testing. Following [24], we set a smaller side to $S$ and isotropically resize the other side. Here we use two scales of images, including $S = 256$ and 320, and perform

average pooling over the topmost layer of FCN. Finally, the multi-scale feature is computed by averaging one CNN feature and two FCN features. Notably, the multi-scale testing can be used for both single Convnet and multi-Convnet aggregation. We also test more scales such as $S = 384, 512$, but no significant improvement is obtained.

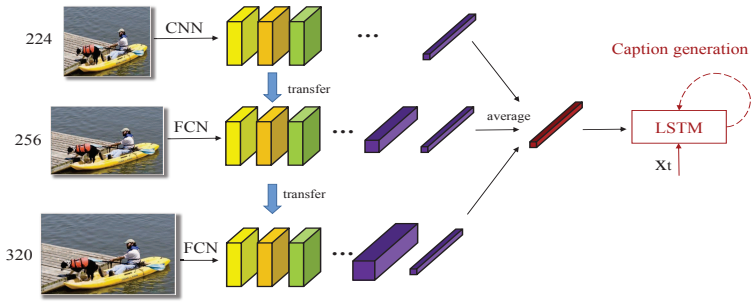



**Figure 6.5:** The pipeline of multi-scale testing approach. Apart from the basic CNN feature, we use two extra scales based on FCNs. We compute the average over three feature vectors and feed it to LSTM units for caption generation.

## 6.4 Experiments

In this section, we evaluate our approach on the well-known MS COCO dataset [305]. MS COCO consists of 82783 training images, 40504 validation images and 40775 testing images. Each image is annotated by at least five human-written captions. Following most recent works [266, 302, 306, 312] , we use 5000 images as validation set to tune hyper-parameters, and another 5000 images as test set to report results. We use the vocabulary dictionary in [266] (containing 8800 words). This dictionary is used to encode the input sequence of words. We implemented our approach based on the Caffe framework [218].

### 6.4.1 Evaluation Configuration

We evaluate our approaches on two tasks: caption generation and image-sentence retrieval. For caption generation task, we evaluate our method with four metrics:

BLEU [313], METEOR [314], ROUGE-L [315] and CIDEr [316]. For image-sentence retrieval task, we divide it into two parts: image-to-sentence retrieval and sentence-to-image retrieval. Following previous works [266, 302], we adopt the evaluation metrics: R@K and Med r. All metrics are computed with the MS COCO evaluation code [317].

We denote the three single Convnets as **SL-Net, ML-Net and MA-Net**. **MA_ML-Net** is the combination of MA-Net and ML-Net, and **MA_ML_SL-Net** indicates the method that aggregates the three Convnets.

We utilize BeamSearch when generating the sentences: iteratively consider the $k$ best sentences up to timestep $t$ when generating sentences of timestep $t+1$. Most of our results use a beam search of size 1 for fast evaluating. For fair comparison with the state-of-the-art, we give the results by using a beam of size 5.

### 6.4.2 Results on Caption Generation

We evaluate our approach on caption generation with 5000 test images. Table 6.1 shows the single-scale and multi-scale testing of the three Convnets. We list the dimension of the feature since it is closely related with the number of LSTM parameters. It is interesting to see that, SL-Net, which utilizes the largest dimension feature, performs the worst among the three Convnets. This demonstrates that increasing the number of system parameters would not necessarily improve the performance.

For single-scale testing, ML-Net brings about 1% boost over the SL-Net for most evaluation metrics. This improvement is marginal compared to the MA-Net, which outperforms the SL-Net significantly over all the evaluation metrics. Notably, the increase of CIDEr reaches 0.093, from 0.703 to 0.796. On the other hand, the multi-scale testing using FCN shows considerable improvement over the corresponding single-scale testing, with the same feature dimension. This is promising, especially considering the high efficiency of FCN.

In addition to evaluating the three Convnets individually, we also explore the effect of aggregating the Convnets, as shown in Table 6.2. We build the multi-

**Table 6.1:** MS COCO results on caption generation by comparing three Convnets. Both single-scale and multi-scale testing are shown. Here we use a beam search of size 1.

| Method | Dim | B-1 | B-2 | B-3 | B-4 | M | R | C |
|---|---|---|---|---|---|---|---|---|
| **Single-scale Testing:** | | | | | | | | |
| SL-Net | 1000 | 0.651 | 0.474 | 0.333 | 0.229 | 0.214 | 0.483 | 0.703 |
| ML-Net | 80 | 0.664 | 0.487 | 0.345 | 0.241 | 0.213 | 0.487 | 0.717 |
| MA-Net | 300 | 0.686 | 0.516 | 0.374 | 0.266 | 0.228 | 0.506 | 0.796 |
| **Multi-scale Testing:** | | | | | | | | |
| SL-Net | 1000 | 0.666 | 0.489 | 0.345 | 0.239 | 0.219 | 0.489 | 0.735 |
| ML-Net | 80 | 0.679 | 0.496 | 0.351 | 0.245 | 0.219 | 0.49 | 0.75 |
| MA-Net | 300 | 0.697 | 0.528 | 0.384 | 0.274 | 0.231 | 0.511 | 0.81 |

Convnet based on MA-Net since it is the best individual Convnet. Overall, both MA_ML-Net and MA_MC_SC-Net perform better than the individual MA-Net, indicating that aggregating the Convnet is beneficial for the caption generation. This is reasonable given the fact that different Convnets would learn different contents, and aggregating them generally lead to a more comprehensive prediction. Furthermore, we also evaluate the multi-scale performance using FCN. Similarly, the multi-scale scheme improves the accuracy of the evaluation metric remarkably. Finally, MA_MC_SC-Net can yield a quite competitive result, such as 0.704 B-1 and 0.846 CIDEr.

**Table 6.2:** MS COCO results on caption generation by multi-Convnet aggregation. The results are based on BLEU, METEOR (M), ROUGE-L (R) and CIDEr (C) metrics. Here we use a beam search of size 1.

| Method | B-1 | B-2 | B-3 | B-4 | M | R | C |
|---|---|---|---|---|---|---|---|
| **Single-scale Testing:** | | | | | | | |
| MA-Net | 0.686 | 0.516 | 0.374 | 0.266 | 0.228 | 0.506 | 0.796 |
| MA_ML-Net | 0.687 | 0.519 | 0.376 | 0.268 | 0.229 | 0.507 | 0.797 |
| MA_ML_SL-Net | 0.688 | 0.52 | 0.379 | 0.27 | 0.229 | 0.507 | 0.803 |
| **Multi-scale Testing:** | | | | | | | |
| MA-Net | 0.697 | 0.528 | 0.384 | 0.274 | 0.231 | 0.511 | 0.81 |
| MA_ML-Net | 0.703 | 0.537 | 0.393 | 0.282 | 0.234 | 0.516 | 0.846 |
| MA_ML_SL-Net | 0.704 | 0.54 | 0.398 | 0.287 | 0.236 | 0.519 | 0.848 |

**Comparison with the state-of-the-art** We compare our MA_MC_SC-Net

result with current state-of-the-art methods in Table 6.3. It can be seen that our results delivered better results than most existing methods. Compared to [303], our method obtained the same result on Bleu-1 with the soft-attention model, slightly worse than the more sophisticated hard-attention model. But for all the other evaluation metrics, our method achieved considerably better results. Similar situation comes with [268], with which we also achieved overall competitive performance. It is worthwhile to say that, our method is not inherently conflicted with these methods, and we can incorporate them together for a better achievement. Note that [312] further improved their results by extracting, clustering and selecting a large number of region proposals. Therefore, their great gains are achieved at the expense of algorithm complexity. In contrast, benefited from the high efficiency of FCN, our multi-scale testing strategy brings negligible extra cost compared to the single-scale testing. We argue that a sophisticated region detection approach [44] is also applicable to our system, but it is out of the scope of this work. Figure 6.6 shows some captioning examples.

**Table 6.3:** Comparison with current state-of-the-art on MS COCO caption generation. Here we use a beam search of size 5.

| Method | B-1 | B-2 | B-3 | B-4 | M | C |
|---|---|---|---|---|---|---|
| Karpathy et al. [302] | 0.625 | 0.450 | 0.321 | 0.230 | 0.195 | 0.66 |
| mRNN [304] | 0.670 | 0.490 | 0.350 | 0.250 | - | - |
| NIC [267] | - | - | - | 0.277 | 0.237 | 0.855 |
| LRCN [266] | 0.669 | 0.489 | 0.349 | 0.249 | - | - |
| gLSTM [307] | 0.670 | 0.491 | 0.358 | 0.264 | 0.227 | 0.813 |
| Bi-LSTM [306] | 0.672 | 0.492 | 0.352 | 0.244 | 0.208 | 0.666 |
| VNet-ft-LSTM [312] | 0.680 | 0.500 | 0.370 | 0.250 | 0.220 | 0.730 |
| Soft-Attention [303] | **0.707** | 0.492 | 0.344 | 0.243 | 0.239 | - |
| Hard-Attention [303] | **0.718** | 0.504 | 0.357 | 0.250 | 0.230 | - |
| Jin et al. [308] | 0.697 | 0.519 | 0.381 | 0.282 | 0.235 | 0.838 |
| ATT-FCN [268] | 0.709 | 0.537 | 0.402 | **0.304** | **0.243** | - |
| Ours | 0.707 | **0.548** | **0.410** | **0.304** | 0.238 | **0.895** |


**Figure 6.6:** The caption generation results for some MS COCO examples by our MA_MC_SC-Net method. We show both the positive and negative examples.

### 6.4.3 Results on Image-sentence Retrieval

We report the image-to-sentence and sentence-to-image results in Table 6.4. There are 5000 test images and 25,000 captions in total. Overall, MA_MC_SC-Net outperforms other state-of-the-art works on both R@K and Med r.

**Table 6.4:** Image-sentence retrieval results on MS COCO dataset. R@K: higher is better; Med r: lower is better.

| | Image to Sentence | | | | Sentence to Image | | | |
|---|---|---|---|---|---|---|---|---|
| Method | R@1 | R@5 | R@10 | Med r | R@1 | R@5 | R@10 | Med r |
| Karpathy et al. [302] | 16.5 | 39.2 | 52.0 | 9.0 | 10.7 | 29.6 | 42.2 | 14.0 |
| Bi-LSTM [306] | 16.6 | 39.4 | 52.4 | 9.0 | 11.6 | 30.9 | 43.4 | 13.0 |
| Ours | **16.9** | **39.8** | **53.1** | **8.0** | **12.4** | **31.5** | **44.0** | **12.0** |

## 6.5 Conclusion

In this work, we studied the effects of Convnets for the image captioning task. We employed three Convnets based on single-label, multi-label, multi-attribute classification. In addition, we integrated the three Convnets for an richer aggregation feature. During the test stage, we employed an efficient multi-scale augmentation approach. Experiments on MS COCO dataset demonstrated that our approach achieved competitive results for both caption generation and image-sentence retrieval as compared to the state-of-the-art. In the future work, we will strive to make use of the attention mechanism.

# Chapter 7

# Conclusions

## 7.1 Conclusions

In this thesis, we explored and designed deep learning algorithms for better image understanding. The topic of image understanding has long been an active research field, and it aims to visualize and understand the image content in a way that is consistent with human perception. To this end, there are many related tasks, such as image classification, object detection, image retrieval and image captioning to name a few. While all these tasks may seem disjoint, developing a good image representation is essential for all of them.

In Chapter 2, we presented a comprehensive review of the developments of various deep learning algorithms. This chapter is intended to be useful for general neural computing, computer vision and multimedia researchers who are interested in the state-of-the-art in deep learning in computer vision. Generally, the deep learning methods can be divided into four categories according to the basic method they are derived from: Convolutional Neural Networks (CNN), Restricted Boltzmann Machines (RBM), Autoencoder and Sparse Coding. Among these four categories, CNN is the most commonly used for the computer vision area and also the basis of the work in this thesis.

Chapter 3 presents an effective scheme to achieve image features (referred to

as PPC). PPC is derived from the fully-connected CNN activations (referred to as CNN). It aims to learn more information from the image and proposes to extract CNN features from multiple spatial sub-regions, and aggregates the multiple CNN together. Without increasing the complexity during the test phase, the feature is further reduced to the same dimension with CNN (i.e. 4096-D) using PCA. Although it is straightforward to achieve, PPC consistantly delivers better performance than the commonly-used CNN, and has the potential to be useful for many tasks.

Initially, researchers have focused on employing CNN activations based on the fully-connected layers. However, current research studies are giving increased attention to the convolutional layers, since they can preserve the spatial information and contain rich semantic information. A common usage of the convolutional activations is to encode them with the Bag-of-Words (BoW) variants, such as VLAD and Fisher Vector. This pipeline not only preserves high discrimination of the CNN activations, but also incorporates the 'bag' conception to improve the invariance property to scale changes, location changes and occlusions. Motivated by this pipeline, Chapter 4 proposes a novel method to incorporate the CNN feature with the BoW framework. In contrast to the common practice, we do not explicitly generate the codebook, and extensively assign the features to the generated visual words according to the similarity. Instead, we take the feature maps as the 'surrogate' pars, and take the activation values as the assignment strengths for these surrogate parts. As a consequence, our novel feature, i.e. BoSP, is much easier to compute, and has a significantly lower dimension than the common usage of 'CNN+BoW'.

Aside from the traditional image classification task, Chapter 5 suggests addressing the hierarchical image classification task, by incorporating the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). This task is intended to generate multiple image labels in a coarse-to-fine pattern, and thus can provide a better understanding of the categories, especially the fine-grained categories. In addition to addressing the hierarchical image classification task, the CNN-RNN paradigm also has the following potential advantages: (1) It can

improve the traditional leaf-level classification performance by exploiting the relationship between hierarchical labels; (2) It can be built on top of any CNN architecture which is primarily designed for leaf-level classification. Accordingly, we built a high-performance baseline network, i.e. wider-ResNet, based on which the CNN-RNN paradigm achieves remarkably better performance than the state-of-the-art on CIFAR-100. (3) It can enhance the image classification performance when part of the training data is only annotated with coarse labels. This provides a promising direction for weakly supervised learning.

The application of CNN-RNN paradigm is not limited to the image classification task. A more common usage is for the image captioning task. As is similar with the image classification, the captioning performance is also closely dependent on the discriminative capacity of CNNs. In Chapter 6, we investigate the effects of different Convnets on image captioning, i.e. single-label Convnets, multi-label Convnets and multi-attribute Convnets. Since the three Convnets focus on different visual contents in one image, we propose aggregating them together to generate a richer visual representation.

## 7.2 Research Limitations

Although our research has reached its aims, we cannot neglect its limitations and weaknesses.

First, from the general point of view, deep learning is often considered as a black box. It can generate relevant results for the given input, but it is not clear what the final learned network means - under what conditions will it work correctly. In addition, the designing and training processes of the deep neural networks may be sensitive: some small architectural and optimization differences may lead to substantial variance in the final result.

Second, it is not clear how well the algorithms and architectures generalize to images from other domains, such as biological images and medical images, since they are established based on the off-the-shelf models which are pretrained on the ImageNet dataset, and ImageNet consists primarily of accurately annotated

natural images. Moreover, we only evaluated our algorithms on the relatively clean benchmarked datasets, and therefore it is difficult to predict how well the methods will work on imagery containing more complexity and noise.

Third, the designing of some algorithms in this thesis has limited theoretic foundation. Take the proposed BoSP feature as an example, it was inspired by the intuition of attempting to use bag-of-words approach on the feature maps from the learned network because bag-of-words had significant success in image understanding based on salient features. However, there was no guarantee that it would work well, nor is there strong theory which would predict the weaknesses of the learned network.

## 7.3 Future Work

In the future, we will extend our work in the following directions:

**Fusing hand-crafted and deep learned features for image representation:** The hand-crafted feature can be seen as a particular form that a human designer thinks can represent the images well. Before the surge of CNN, hand-crafted feature has long been a key component in the competition-winning systems for visual understanding. In the future, we would like to employ the idea of hand-craft features to design the deep networks, in order to make the networks focus more on the important areas.

**Image captioning with grammar supervision:** Image captioning is a new emerging research area which can describe the image with more informative contents, including the objects, actions, relations and etc. In order to generate novel sentences for unseen scenes, most of the current works employ the generative approaches, such as Baby Talk [301] and LRCN [266]. These approaches generate the words one-by-one, and as a consequence, the whole generated sentences may be oddly organised. To obtain sentences that are more consistent with our language, in our future work, we propose to provide grammar supervision during the training of the network.

**Designing more comprehensive CNN models:** CNN models have achieved significant success in various computer vision tasks, including image classification, object detection, image retrieval, image captioning, to name a few. These seemingly disjoint tasks do have some fundamental similarities. For example, Liu et al. [318] proposed to utilize the segmentation annotations to help the edge detection. Oquab et al. [37] took advantage of the object location to improve the image classification performance. While most of these works end up with task-specific CNN models, we assume that the 'real' artificial intelligence should be capable of tackling a broad set of computer vision problems. Therefore, in our future work, we want to exploit the synergy between different visual tasks, and design a universal network that can solve multiple tasks,

# Bibliography


[1] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110

[2] Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Proceedings of European Conference on Computer Vision. (2006) 404–417

[3] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2005) 886–893

[4] Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proceedings of the IEEE International Conference on Computer Vision. (2003) 1470–1477

[5] Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P., Schmid, C.: Aggregating local image descriptors into compact codes. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 1704–1716

[6] Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–8

[7] Sargano, A.B., Angelov, P., Habib, Z.: A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition. Applied Sciences **7** (2017) 110

[8] Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural computation **18** (2006) 1527–1554

[9] He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1026–1034

[10] Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence **39** (2017) 652–663

[11] Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. In: International Conference on Artificial Intelligence and Statistics. (2012) 127–135

[12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. (2013) 3111–3119

[13] Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 3642–3649

[14] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. (2012) 1097–1105

[15] Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing **3** (2014) e2

[16] Bengio, Y., et al.: Learning deep architectures for ai. Foundations and trends® in Machine Learning **2** (2009) 1–127

[17] Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61** (2015) 85–117

[18] Bengio, Y.: Deep learning of representations: Looking forward. In: International Conference on Statistical Language and Speech Processing. (2013) 1–37

[19] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 1798–1828

[20] LeCun, Y.: Learning invariant feature hierarchies. In: Proceedings of European Conference on Computer Vision. (2012) 496–505

[21] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115** (2015) 211–252

[22] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proceedings of European Conference on Computer Vision. (2014) 818–833

[23] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Proceedings of European Conference on Computer Vision. (2014) 346–361

[24] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. (2015)

[25] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1–9

[26] Salakhutdinov, R., Hinton, G.: Deep boltzmann machines. In: International Conference on Artificial Intelligence and Statistics. (2009) 448–455

[27] Ngiam, J., Chen, Z., Koh, P.W., Ng, A.Y.: Learning deep energy models. In: International Conference on Machine Learning. (2011) 1105–1112

[28] Poultney, C., Chopra, S., Cun, Y.L., et al.: Efficient learning of sparse representations with an energy-based model. In: Advances in Neural Information Processing Systems. (2007) 1137–1144

[29] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: International Conference on Machine Learning. (2008) 1096–1103

[30] Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: International Conference on Machine Learning. (2011) 833–840

[31] Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2009) 1794–1801

[32] Gao, S., Tsang, I.W.H., Chia, L.T., Zhao, P.: Local features are not lonely–laplacian sparse coding for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2010) 3555–3561

[33] Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: Advances in Neural Information Processing Systems. (2009) 2223–2231

[34] Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image classification using super-vector coding of local image descriptors. In: Proceedings of European Conference on Computer Vision. (2010) 141–154

[35] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86** (1998) 2278–2324

[36] Zeiler, M.D.: Hierarchical convolutional deep learning in computer vision. PhD thesis, NEW YORK UNIVERSITY (2013)

[37] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 685–694

[38] Lin, M., Chen, Q., Yan, S.: Network in network. In: International Conference on Learning Representations. (2014)

[39] Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: International Conference on Machine Learning. (2010) 111–118

[40] Scherer, D., Müller, A., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. In: International Conference on Artificial Neural Networks. (2010) 92–101

[41] Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: High-performance neural networks for visual object classification. arXiv preprint arXiv:1102.0183 (2011)

[42] Zeiler, M.D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. In: International Conference on Learning Representations. (2013)

[43] Ouyang, W., Luo, P., Zeng, X., Qiu, S., Tian, Y., Li, H., Yang, S., Wang, Z., Xiong, Y., Qian, C., et al.: Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection. arXiv preprint arXiv:1409.3505 (2014)

[44] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 580–587

[45] Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1717–1724

[46] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)

[47] Baldi, P., Sadowski, P.J.: Understanding dropout. In: Advances in Neural Information Processing Systems. (2013) 2814–2822

[48] Ba, J., Frey, B.: Adaptive dropout for training deep neural networks. In: Advances in Neural Information Processing Systems. (2013) 3084–3092

[49] McAllester, D.: A pac-bayesian tutorial with a dropout bound. arXiv preprint arXiv:1307.2118 (2013)

[50] Wager, S., Wang, S., Liang, P.S.: Dropout training as adaptive regularization. In: Advances in Neural Information Processing Systems. (2013) 351–359

[51] Wang, S.I., Manning, C.D.: Fast dropout training. In: International Conference on Machine Learning. (2013) 118–126

[52] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15** (2014) 1929–1958

[53] Warde-Farley, D., Goodfellow, I.J., Courville, A., Bengio, Y.: An empirical analysis of dropout in piecewise linear networks. In: International Conference on Learning Representations. (2014)

[54] Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: International Conference on Machine Learning. (2013) 1058–1066

[55] Howard, A.G.: Some improvements on deep convolutional neural network based image classification. arXiv preprint arXiv:1312.5402 (2013)

[56] Dosovitskiy, A., Springenberg, J.T., Brox, T.: Unsupervised feature learning by augmenting single images. arXiv preprint arXiv:1312.5242 (2013)

[57] Wu, R., Yan, S., Shan, Y., Dang, Q., Sun, G.: Deep image: Scaling up image recognition. arXiv preprint arXiv:1501.02876 (2015)

[58] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S.: Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research **11** (2010) 625–660

[59] He, K., Sun, J.: Convolutional neural networks at constrained time cost. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 5353–5360

[60] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440

[61] Wan, J., Wang, D., Hoi, S.C.H., Wu, P., Zhu, J., Zhang, Y., Li, J.: Deep learning for content-based image retrieval: A comprehensive study. In: Proceedings of the ACM International Conference on Multimedia. (2014) 157–166

[62] Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems. (2014) 3320–3328

[63] Ngiam, J., Chen, Z., Chia, D., Koh, P.W., Le, Q.V., Ng, A.Y.: Tiled convolutional neural networks. In: Advances in Neural Information Processing Systems. (2010) 1279–1287

[64] Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1440–1448

[65] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. (2015) 91–99

[66] Zhu, Y., Urtasun, R., Salakhutdinov, R., Fidler, S.: segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4703–4711

[67] Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1134–1142

[68] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: Proceedings of European Conference on Computer Vision. (2014) 297–312

[69] Zhang, Y., Sohn, K., Villegas, R., Pan, G., Lee, H.: Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 249–258

[70] Yoo, D., Park, S., Lee, J.Y., So Kweon, I.: Multi-scale pyramid pooling for deep convolutional representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop. (2015) 71–80

[71] Xie, S., Tu, Z.: Holistically-nested edge detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1395–1403

[72] Sun, Y., Wang, X., Tang, X.: Deep convolutional network cascade for facial point detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3476–3483

[73] Wang, X., Zhang, L., Lin, L., Liang, Z., Zuo, W.: Deep joint task learning for generic object extraction. In: Advances in Neural Information Processing Systems. (2014) 523–531

[74] Zeng, X., Ouyang, W., Wang, X.: Multi-stage contextual deep learning for pedestrian detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 121–128

[75] Miclut, B.: Committees of deep feedforward networks trained with few data. In: German Conference on Pattern Recognition. (2014) 736–742

[76] Weston, J., Ratle, F., Mobahi, H., Collobert, R.: Deep learning via semi-supervised embedding. In: Neural Networks: Tricks of the Trade. (2012) 639–655

[77] Simonyan, K., Vedaldi, A., Zisserman, A.: Deep fisher networks for large-scale image classification. In: Advances in Neural Information Processing Systems. (2013) 163–171

[78] Chen, Q., Song, Z., Dong, J., Huang, Z., Hua, Y., Yan, S.: Contextualizing object detection and classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **37** (2015) 13–27

[79] Hinton, G.E., Sejnowski, T.J.: Learning and releaming in boltzmann machines. Parallel Distrilmted Processing **1** (1986)

[80] Carreira-Perpinan, M.A., Hinton, G.E.: On contrastive divergence learning. In: International Conference on Artificial Intelligence and Statistics. (2005) 33–40

[81] Hinton, G.E.: A practical guide to training restricted boltzmann machines. In: Neural networks: Tricks of the trade. (2012) 599–619

[82] Cho, K., Raiko, T., Ihler, A.T.: Enhanced gradient and adaptive learning rate for training restricted boltzmann machines. In: International Conference on Machine Learning. (2011) 105–112

[83] Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: International Conference on Machine Learning. (2010) 807–814

[84] Arel, I., Rose, D.C., Karnowski, T.P.: Deep machine learning-a new frontier in artificial intelligence research [research frontier]. IEEE Computational Intelligence Magazine **5** (2010) 13–18

[85] Lee, H., Ekanadham, C., Ng, A.Y.: Sparse deep belief net model for visual area v2. In: Advances in Neural Information Processing Systems. (2008) 873–880

[86] Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: International Conference on Machine Learning. (2009) 609–616

[87] Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Unsupervised learning of hierarchical representations with convolutional deep belief networks. Communications of the ACM **54** (2011) 95–103

[88] Nair, V., Hinton, G.E.: 3d object recognition with deep belief nets. In: Advances in Neural Information Processing Systems. (2009) 1339–1347

[89] Tang, Y., Eliasmith, C.: Deep networks for robust visual recognition. In: International Conference on Machine Learning. (2010) 1055–1062

[90] Huang, G.B., Lee, H., Learned-Miller, E.: Learning hierarchical representations for face verification with convolutional deep belief networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 2518–2525

[91] Younes, L.: On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. Stochastics: An International Journal of Probability and Stochastic Processes **65** (1999) 177–228

[92] Salakhutdinov, R., Larochelle, H.: Efficient learning of deep boltzmann machines. In: International Conference on Artificial Intelligence and Statistics. (2010) 693–700

[93] Salakhutdinov, R.R., Hinton, G.E.: An efficient learning procedure for deep boltzmann machines. Neural computation **24** (2012) 1967–2006

[94] Hinton, G.E., Salakhutdinov, R.R.: A better way to pretrain deep boltzmann machines. In: Advances in Neural Information Processing Systems. (2012) 2447–2455

[95] Cho, K., Raiko, T., Ilin, A., Karhunen, J.: A two-stage pretraining algorithm for deep boltzmann machines. In: International Conference on Artificial Neural Networks, Springer (2013) 106–113

[96] Montavon, G., Müller, K.R.: Deep boltzmann machines and the centering trick. In: Neural Networks: Tricks of the Trade. (2012) 621–637

[97] Goodfellow, I.J., Courville, A., Bengio, Y.: Joint training deep boltzmann machines for classification. arXiv preprint arXiv:1301.3568 (2013)

[98] Goodfellow, I., Mirza, M., Courville, A., Bengio, Y.: Multi-prediction deep boltzmann machines. In: Advances in Neural Information Processing Systems. (2013) 548–556

[99] Elfwing, S., Uchibe, E., Doya, K.: Expected energy-based restricted boltzmann machine for classification. Neural Networks **64** (2015) 29–38

[100] Eslami, S.A., Heess, N., Williams, C.K., Winn, J.: The shape boltzmann machine: a strong model of object shape. International Journal of Computer Vision **107** (2014) 155–176

[101] Kae, A., Sohn, K., Lee, H., Learned-Miller, E.: Augmenting crfs with boltzmann machine shape priors for image labeling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 2019–2026

[102] Dahl, G., Mohamed, A.r., Hinton, G.E., et al.: Phone recognition with the mean-covariance restricted boltzmann machine. In: Advances in Neural Information Processing Systems. (2010) 469–477

[103] Liou, C.Y., Cheng, W.C., Liou, J.W., Liou, D.R.: Autoencoder for words. Neurocomputing **139** (2014) 84–96

[104] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science **313** (2006) 504–507

[105] Zhang, J., Shan, S., Kan, M., Chen, X.: Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment. In: Proceedings of European Conference on Computer Vision. (2014) 1–16

[106] Jiang, X., Zhang, Y., Zhang, W., Xiao, X.: A novel sparse auto-encoder for deep unsupervised learning. In: International Conference on Advanced Computational Intelligence. (2013) 256–261

[107] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A.: Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research **11** (2010) 3371–3408

[108] Goroshin, R., LeCun, Y.: Saturating auto-encoders. arXiv preprint arXiv:1301.3577 (2013)

[109] Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: International Conference on Artificial Neural Networks. (2011) 52–59

[110] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A.: Spatio-temporal convolutional sparse auto-encoder for sequence classification. In: British Machine Vision Conference. (2012)

[111] Leng, B., Guo, S., Zhang, X., Xiong, Z.: 3d object retrieval with stacked local convolutional autoencoder. Signal Processing **112** (2015) 119–128

[112] Konda, K., Memisevic, R., Krueger, D.: Zero-bias autoencoders and the benefits of co-adapting features. In: International Conference on Learning Representations. (2015)

[113] Goodfellow, I., Lee, H., Le, Q.V., Saxe, A., Ng, A.Y.: Measuring invariances in deep networks. In: Advances in Neural Information Processing Systems. (2009) 646–654

[114] Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Le, Q.V., Ng, A.Y.: On optimization methods for deep learning. In: International Conference on Machine Learning. (2011) 265–272

[115] Zou, W.Y., Ng, A.Y., Yu, K.: Unsupervised learning of visual invariance with temporal coherence. In: Advances in Neural Information Processing Systems Workshop. (2011)

[116] Simoncelli, E.P.: Statistical modeling of photographic images. In: Handbook of image and video processing. (2005)

[117] Le, Q.V.: Building high-level features using large scale unsupervised learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing. (2013) 8595–8598

[118] Alain, G., Bengio, Y.: What regularized auto-encoders learn from the data-generating distribution. Journal of Machine Learning Research **15** (2014) 3563–3593

[119] Mesnil, G., Dauphin, Y., Glorot, X., Rifai, S., Bengio, Y., Goodfellow, I.J., Lavoie, E., Muller, X., Desjardins, G., Warde-Farley, D., et al.: Unsupervised and transfer learning challenge: a deep learning approach. In: International Conference on Machine Learning Workshop. (2012) 97–110

[120] Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research **37** (1997) 3311–3325

[121] Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data. In: International Conference on Machine Learning. (2007) 759–766

[122] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2010) 3360–3367

[123] Donoho, D.L.: For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. Communications on pure and applied mathematics **59** (2006) 797–829

[124] Censor, Y., Zenios, S.A.: Parallel optimization: Theory, algorithms, and applications. Oxford University Press on Demand (1997)

[125] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Cognitive modeling **5** (1988) 1

[126] Lee, H., Battle, A., Raina, R., Ng, A.Y.: Efficient sparse coding algorithms. In: Advances in Neural Information Processing Systems. (2007)

[127] Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: International Conference on Machine Learning. (2009) 689–696

[128] Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research **11** (2010) 19–60

[129] Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al.: Pathwise coordinate optimization. The Annals of Applied Statistics **1** (2007) 302–332

[130] Gregor, K., LeCun, Y.: Learning fast approximations of sparse coding. In: International Conference on Machine Learning. (2010) 399–406

[131] Chambolle, A., De Vore, R.A., Lee, N.Y., Lucier, B.J.: Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. IEEE Transactions on Image Processing **7** (1998) 319–335

[132] Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In: IEEE International Conference on Acoustics, Speech and Signal Processing. (2009) 693–696

[133] Kavukcuoglu, K., Ranzato, M., LeCun, Y.: Fast inference in sparse coding algorithms with applications to object recognition. arXiv preprint arXiv:1010.3467 (2010)

[134] Balasubramanian, K., Yu, K., Lebanon, G.: Smooth sparse coding via marginal regression for learning sparse representations. In: International Conference on Machine Learning. (2013) 289–297

[135] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2006) 2169–2178

[136] Coates, A., Ng, A.Y.: The importance of encoding versus training with sparse coding and vector quantization. In: International Conference on Machine Learning. (2011) 921–928

[137] Gao, S., Tsang, I.W.H., Chia, L.T.: Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 92–104

[138] Yu, K., Lin, Y., Lafferty, J.: Learning image representations from the pixel level via hierarchical sparse coding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2011) 1713–1720

[139] Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2010) 2528–2535

[140] Zeiler, M.D., Taylor, G.W., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: Proceedings of the IEEE International Conference on Computer Vision. (2011) 2018–2025

[141] Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Cao, L., Huang, T.: Large-scale image classification: fast feature extraction and svm training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2011) 1689–1696

[142] He, Y., Kavukcuoglu, K., Wang, Y., Szlam, A., Qi, Y.: Unsupervised feature learning by deep sparse coding. In: Proceedings of the 2014 SIAM International Conference on Data Mining. (2014) 902–910

[143] Master, S.: Large scale object detection. PhD thesis, Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University (2014)

[144] Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Proceedings of European Conference on Computer Vision Workshop. (2004)

[145] Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational Learning Theory. (1992) 144–152

[146] Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: Proceedings of the IEEE International Conference on Computer Vision. (2009) 32–39

[147] Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Proceedings of European Conference on Computer Vision. (2010) 143–156

[148] Jaakkola, T.S., Haussler, D., et al.: Exploiting generative models in discriminative classifiers. In: Advances in Neural Information Processing Systems. (1999) 487–493

[149] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2009) 248–255

[150] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. In: International Conference on Learning Representations. (2014)

[151] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)

[152] Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence **32** (2010) 1627–1645

[153] Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Advances in Neural Information Processing Systems. (2013) 2553–2561

[154] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 2147–2154

[155] Ren, S., He, K., Girshick, R., Zhang, X., Sun, J.: Object detection networks on convolutional feature maps. IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)

[156] Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International Journal of Computer Vision **104** (2013) 154–171

[157] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 779–788

[158] Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. IEEE Transactions on Pattern Analysis and Machine Intelligence **34** (2012) 2189–2202

[159] Endres, I., Hoiem, D.: Category independent object proposals. In: Proceedings of European Conference on Computer Vision. (2010) 575–588

[160] Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: Bing: Binarized normed gradients for objectness estimation at 300fps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3286–3293

[161] Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: Proceedings of European Conference on Computer Vision. (2014) 391–405

[162] Hosang, J., Benenson, R., Schiele, B.: How good are detection proposals, really? In: British Machine Vision Conference. (2014)

[163] Dai, Q., Hoiem, D.: Learning to localize detected objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 3322–3329

[164] Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: Proceedings of European Conference on Computer Vision. (2012) 340–353

[165] Dong, J., Chen, Q., Yan, S., Yuille, A.: Towards unified object detection and semantic segmentation. In: Proceedings of European Conference on Computer Vision. (2014) 299–314

[166] Hoffman, J., Guadarrama, S., Tzeng, E.S., Hu, R., Donahue, J., Girshick, R., Darrell, T., Saenko, K.: Lsda: Large scale detection through adaptation. In: Advances in Neural Information Processing Systems. (2014) 3536–3544

[167] Hoffman, J., Guadarrama, S., Tzeng, E., Donahue, J., Girshick, R., Darrell, T., Saenko, K.: From large-scale object classifiers to large-scale object detectors: An adaptation approach. In: Advances in Neural Information Processing Systems. (2014)

[168] Zhou, B., Jagadeesh, V., Piramuthu, R.: Conceptlearner: Discovering visual concepts from weakly labeled image collections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1492–1500

[169] Liang, X., Liu, S., Wei, Y., Liu, L., Lin, L., Yan, S.: Towards computational baby learning: A weakly-supervised approach for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 999–1007

[170] Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: Proceedings of European Conference on Computer Vision. (2014) 392–407

[171] Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop. (2014) 806–813

[172] Sun, S., Zhou, W., Li, H., Tian, Q.: Search by detection: Object-level feature for image retrieval. In: Proceedings of International Conference on Internet Multimedia Computing and Service. (2014) 46

[173] Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Proceedings of European Conference on Computer Vision. (2014) 584–599

[174] Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: International Conference on Learning Representations. (2015)

[175] Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3992–4000

[176] Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 447–456

[177] Lin, G., Shen, C., van den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 3194–3203

[178] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1529–1537

[179] Papandreou, G., Chen, L.C., Murphy, K., Yuille, A.L.: Weakly-and semi-supervised learning of a dcnn for semantic image segmentation. arXiv preprint arXiv:1502.02734 (2015)

[180] Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1635–1643

[181] Jain, A., Tompson, J., LeCun, Y., Bregler, C.: Modeep: A deep learning framework using motion features for human pose estimation. In: Asian Conference on Computer Vision. (2014) 302–315

[182] Pfister, T., Simonyan, K., Charles, J., Zisserman, A.: Deep convolutional neural networks for efficient pose estimation in gesture videos. In: Asian Conference on Computer Vision. (2014) 538–552

[183] Pfister, T., Charles, J., Zisserman, A.: Flowing convnets for human pose estimation in videos. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1913–1921

[184] Yu, J., Guo, Y., Tao, D., Wan, J.: Human pose recovery by supervised spectral embedding. Neurocomputing **166** (2015) 301–308

[185] Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: Advances in Neural Information Processing Systems. (2014) 1736–1744

[186] Jain, A., Tompson, J., Andriluka, M., Taylor, G.W., Bregler, C.: Learning human pose estimation features with convolutional networks. In: International Conference on Learning Representations. (2014)

[187] Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems. (2014) 1799–1807

[188] Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 648–656

[189] Ouyang, W., Chu, X., Wang, X.: Multi-source deep learning for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 2329–2336

[190] Fan, X., Zheng, K., Lin, Y., Wang, S.: Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1347–1355

[191] Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4733–4742

[192] Huang, C.H., Boyer, E., Ilic, S.: Robust human body shape and pose tracking. In: 3DTV-Conference, 2013 International Conference on. (2013) 287–294

[193] Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision **61** (2005) 55–79

[194] Tian, Y., Zitnick, C.L., Narasimhan, S.G.: Exploring the spatial hierarchy of mixture models for human pose estimation. In: Proceedings of European Conference on Computer Vision. (2012) 256–269

[195] Wang, F., Li, Y.: Beyond physical connections: Tree models in human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 596–603

[196] Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 588–595

[197] Dantone, M., Gall, J., Leistner, C., Van Gool, L.: Human pose estimation using body parts dependent joint regressors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3041–3048

[198] Sapp, B., Taskar, B.: Modec: Multimodal decomposable models for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 3674–3681

[199] Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: British Machine Vision Conference. (2010)

[200] Eichner, M., Marin-Jimenez, M., Zisserman, A., Ferrari, V.: 2d articulated human pose estimation and retrieval in (almost) unconstrained still images. International Journal of Computer Vision **99** (2012) 190–214

[201] Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 1653–1660

[202] Chu, J.L., Krzyżak, A.: Analysis of feature maps selection in supervised learning using convolutional neural networks. In: Canadian Conference on Artificial Intelligence. (2014) 59–70

[203] Yu, W., Yang, K., Bai, Y., Yao, H., Rui, Y.: Visualizing and comparing convolutional neural networks. arXiv preprint arXiv:1412.6631 (2014)

[204] Agrawal, P., Girshick, R., Malik, J.: Analyzing the performance of multilayer neural networks for object recognition. In: Proceedings of European Conference on Computer Vision. (2014) 329–344

[205] Cadieu, C.F., Hong, H., Yamins, D.L., Pinto, N., Ardila, D., Solomon, E.A., Majaj, N.J., DiCarlo, J.J.: Deep neural networks rival the representation of primate it cortex for core visual object recognition. PLoS Comput Biol **10** (2014) e1003963

[206] Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 427–436

[207] Firat, O., Aksan, E., Oztekin, I., Vural, F.T.Y.: Learning deep temporal representations for brain decoding. arXiv preprint arXiv:1412.7522 (2014)

[208] Chen, X., Shrivastava, A., Gupta, A.: Neil: Extracting visual knowledge from web data. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1409–1416

[209] Divvala, S.K., Farhadi, A., Guestrin, C.: Learning everything about anything: Webly-supervised visual concept learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3270–3277

[210] Song, H.O., Lee, Y.J., Jegelka, S., Darrell, T.: Weakly-supervised discovery of visual pattern configurations. In: Advances in Neural Information Processing Systems. (2014) 1637–1645

[211] Li, H., Zhao, R., Wang, X.: Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification. arXiv preprint arXiv:1412.4526 (2014)

[212] Ren, J.S., Xu, L.: On vectorization of deep convolutional neural networks for vision tasks. In: AAAI Conference on Artificial Intelligence. (2015)

[213] Liu, Y., Guo, Y., Wu, S., Lew, M.S.: Deepindex for accurate and efficient image retrieval. In: Proceedings of the ACM International Conference on Multimedia Retrieval. (2015) 43–50

[214] Zheng, L., Wang, S., He, F., Tian, Q.: Seeing the big picture: Deep embedding with contextual evidences. arXiv preprint arXiv:1406.0132 (2014)

[215] Yan, Z., Jagadeesh, V., DeCoste, D., Di, W., Piramuthu, R.: Hd-cnn: Hierarchical deep convolutional neural network for image classification. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2740–2748

[216] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: Advances in Neural Information Processing Systems. (2014) 487–495

[217] Koskela, M., Laaksonen, J.: Convolutional network features for scene recognition. In: Proceedings of the ACM International Conference on Multimedia. (2014) 1169–1172

[218] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. (2014) 675–678

[219] Jolliffe, I.: Principal component analysis. Wiley Online Library (2002)

[220] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding **106** (2007) 59–70

[221] Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2009) 413–420

[222] Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Proceedings of European Conference on Computer Vision. (2008) 304–317

[223] Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27

[224] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2007) 1–8

[225] Seber, G.A.: Multivariate observations. Volume 252. John Wiley & Sons (2009)

[226] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: A review. Neurocomputing **187** (2016) 27–48

[227] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 770–778

[228] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: Decaf: A deep convolutional activation feature for generic visual recognition. In: International Conference on Machine Learning. (2014) 647–655

[229] Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: Cnn features off-the-shelf: an astounding baseline for recognition. In: Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 806–813

[230] Xiu-Shen, W., Bin-Bin, G., Jianxin, W.: Deep spatial pyramid ensemble for cultural event recognition. In: Proceedings of the IEEE International Conference on Computer Vision Workshop. (2015)

[231] Liu, L., Shen, C., Hengel, A.v.d.: The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4749–4757

[232] Babenko, A., Lempitsky, V.: Aggregating local deep features for image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1269–1277

[233] Cimpoi, M., Maji, S., Vedaldi, A.: Deep filter banks for texture recognition and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3828–3836

[234] Liu, L., Shen, C., van den Hengel, A.: Cross-convolutional-layer pooling for image recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2016)

[235] Ng, J., Yang, F., Davis, L.: Exploiting local features from deep networks for image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop. (2015) 53–61

[236] Cimpoi, M., Maji, S., Kokkinos, I., Vedaldi, A.: Deep filter banks for texture recognition, description, and segmentation. International Journal of Computer Vision **118** (2016) 65–94

[237] Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A., et al.: Sun database: Large-scale scene recognition from abbey to zoo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2010) 3485–3492

[238] Parizi, S.N., Vedaldi, A., Zisserman, A., Felzenszwalb, P.: Automatic discovery and optimization of parts for image classification. In: International Conference on Learning Representations. (2015)

[239] Singh, S., Gupta, A., Efros, A.: Unsupervised discovery of mid-level discriminative patches. In: Proceedings of European Conference on Computer Vision. (2012) 73–86

[240] Juneja, M., Vedaldi, A., Jawahar, C., Zisserman, A.: Blocks that shout: Distinctive parts for scene classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2013) 923–930

[241] Kulkarni, P., Jurie, F., Zepeda, J., Pérez, P., Chevallier, L.: Spleap: Soft pooling of learned parts for image classification. In: Proceedings of European Conference on Computer Vision. (2016) 329–345

[242] Wu, J., Gao, B.B., Liu, G.: Representing sets of instances for visual recognition. In: AAAI Conference on Artificial Intelligence. (2016)

[243] Liu, L., Wang, L., Liu, X.: In defense of soft-assignment coding. In: Proceedings of the IEEE International Conference on Computer Vision. (2011) 2486–2493

[244] Fernando, B., Fromont, E., Muselet, D., Sebban, M.: Discriminative feature fusion for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2012) 3434–3441

[245] Minka, T.P.: A comparison of numerical optimizers for logistic regression. Technical report (2003)

[246] Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9** (2008) 2579–2605

[247] Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics & Image Processing. (2008) 722–729

[248] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. Journal of machine learning research **9** (2008) 1871–1874

[249] Swain, M.J., Ballard, D.H.: Color indexing. International Journal of Computer Vision **7** (1991) 11–32

[250] Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., Oliva, A.: Places: An image database for deep scene understanding. arXiv preprint arXiv:1610.02055 (2016)

[251] Xie, L., Tian, Q., Hong, R., Zhang, B.: Image classification and retrieval are one. In: Proceedings of the ACM International Conference on Multimedia Retrieval. (2015) 3–10

[252] Yang, S., Ramanan, D.: Multi-scale recognition with dag-cnns. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1215–1223

[253] Sharma, G., Schiele, B.: Scalable nonlinear embeddings for semantic category-based image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1296–1304

[254] Li, Y., Liu, L., Shen, C., van den Hengel, A.: Mining mid-level visual patterns with deep cnn activations. International Journal of Computer Vision (2016) 1–21

[255] Herranz, L., Jiang, S., Li, X.: Scene recognition with cnns: objects, scales and dataset bias. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 571–579

[256] Kim, Y.D., Jang, T., Han, B., Choi, S.: Learning to select pre-trained deep representations with bayesian evidence framework. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 5318–5326

[257] Wei, X.S., Luo, J.H., Wu, J., Zhou, Z.H.: Selective convolutional descriptor aggregation for fine-grained image retrieval. IEEE Transactions on Image Processing (2017)

[258] Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–8

[259] Guo, Y., Lew, M.S.: Bag of surrogate parts: one inherent feature of deep cnns. In: British Machine Vision Conference. (2016)

[260] Shirahama, K., Grzegorzek, M.: Towards large-scale multimedia retrieval enriched by knowledge about human interpretation. Multimedia Tools and Applications **75** (2016) 297–331

[261] Cao, L., Gao, L., Song, J., Shen, F., Wang, Y.: Multiple hierarchical deep hashing for large scale image retrieval. Multimedia Tools and Applications (2017) 1–14

[262] Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference. (2016)

[263] Liu, Y., Guo, Y., Lew, M.S.: On the exploration of convolutional fusion networks for visual recognition. In: International Conference on Multimedia Modeling. (2017) 277–289

[264] Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio, S., Li, Y., Neven, H., Adam, H.: Large-scale object classification using label relation graphs. In: Proceedings of European Conference on Computer Vision. (2014) 48–64

[265] Ristin, M., Gall, J., Guillaumin, M., Van Gool, L.: From categories to subcategories: large-scale image classification with partial class label refinement. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 231–239

[266] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 2625–2634

[267] Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3156–3164

[268] You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4651–4659

[269] Zuo, Z., Shuai, B., Wang, G., Liu, X., Wang, X., Wang, B., Chen, Y.: Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop. (2015) 18–26

[270] Liang, M., Hu, X.: Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3367–3375

[271] Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: Cnn-rnn: A unified framework for multi-label image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2285–2294

[272] Visin, F., Kastner, K., Cho, K., Matteucci, M., Courville, A., Bengio, Y.: Renet: A recurrent neural network based alternative to convolutional networks. arXiv preprint arXiv:1505.00393 (2015)

[273] Yan, G., Wang, Y., Liao, Z.: Lstm for image annotation with relative visual importance. In: British Machine Vision Conference. (2016)

[274] Salakhutdinov, R., Torralba, A., Tenenbaum, J.: Learning to share visual appearance for multiclass object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2011) 1481–1488

[275] Murdock, C., Li, Z., Zhou, H., Duerig, T.: Blockout: Dynamic model selection for hierarchical deep networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 2583–2591

[276] Elman, J.L.: Finding structure in time. Cognitive science **14** (1990) 179–211

[277] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9** (1997) 1735–1780

[278] Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto (2009)

[279] He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Proceedings of European Conference on Computer Vision. (2016) 630–645

[280] Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: International Conference on Learning Representations. (2015)

[281] Lee, C.Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z.: Deeply-supervised nets. In: International Conference on Artificial Intelligence and Statistics. (2015) 562–570

[282] Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. In: International Conference on Learning Representations Workshop. (2015)

[283] Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. In: International Conference on Learning Representations Workshop. (2015)

[284] Agostinelli, F., Hoffman, M., Sadowski, P., Baldi, P.: Learning activation functions to improve deep neural networks. In: International Conference on Learning Representations Workshop. (2015)

[285] Jin, X., Xu, C., Feng, J., Wei, Y., Xiong, J., Yan, S.: Deep learning with s-shaped rectified linear activation units. In: AAAI Conference on Artificial Intelligence. (2016)

[286] Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M.M.A., Prabhat, M., Adams, R.P.: Scalable bayesian optimization using deep neural networks. In: International Conference on Machine Learning. (2015) 2171–2180

[287] Mishkin, D., Matas, J.: All you need is a good init. In: International Conference on Learning Representations. (2016)

[288] Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). In: International Conference on Learning Representations. (2016)

[289] Li, H., Ouyang, W., Wang, X.: Multi-bias non-linear activation in deep neural networks. In: International Conference on Machine Learning. (2016)

[290] Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: Proceedings of European Conference on Computer Vision. (2016) 646–661

[291] Targ, S., Almeida, D., Lyman, K.: Resnet in resnet: generalizing residual architectures. In: International Conference on Learning Representations Workshop. (2016)

[292] Larsson, G., Maire, M., Shakhnarovich, G.: Fractalnet: Ultra-deep neural networks without residuals. In: International Conference on Learning Representations. (2017)

[293] Singh, S., Hoiem, D., Forsyth, D.: Swapout: Learning an ensemble of deep architectures. In: Advances in Neural Information Processing Systems. (2016) 28–36

[294] Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38** (1995) 39–41

[295] Mensink, T., Verbeek, J., Perronnin, F., Csurka, G.: Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 2624–2637

[296] Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Good practice in large-scale learning for image classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **36** (2014) 507–520

[297] Ordonez, V., Kulkarni, G., Berg, T.L.: Im2text: Describing images using 1 million captioned photographs. In: Advances in Neural Information Processing Systems. (2011) 1143–1151

[298] Hodosh, M., Young, P., Hockenmaier, J.: Framing image description as a ranking task: Data, models and evaluation metrics. Journal of Artificial Intelligence Research **47** (2013) 853–899

[299] Kuznetsova, P., Ordonez, V., Berg, A.C., Berg, T.L., Choi, Y.: Collective generation of natural image descriptions. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. (2012) 359–368

[300] Mitchell, M., Han, X., Dodge, J., Mensch, A., Goyal, A., Berg, A., Yamaguchi, K., Berg, T., Stratos, K., Daumé III, H.: Midge: Generating image descriptions from computer vision detections. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. (2012) 747–756

[301] Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2011)

[302] Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3128–3137

[303] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation

with visual attention. In: International Conference on Machine Learning. (2015) 2048–2057

[304] Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-rnn). In: International Conference on Learning Representations. (2015)

[305] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Proceedings of European Conference on Computer Vision. (2014) 740–755

[306] Wang, C., Yang, H., Bartz, C., Meinel, C.: Image captioning with deep bidirectional lstms. In: Proceedings of the 2016 ACM on Multimedia Conference. (2016) 988–997

[307] Jia, X., Gavves, E., Fernando, B., Tuytelaars, T.: Guiding the long-short term memory model for image caption generation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2407–2415

[308] Jin, J., Fu, K., Cui, R., Sha, F., Zhang, C.: Aligning where to see and what to tell: image caption with region-based attention and scene factorization. arXiv preprint arXiv:1506.06272 (2015)

[309] Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., et al.: From captions to visual concepts and back. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 1473–1482

[310] Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4565–4574

[311] Yao, L., Ballas, N., Cho, K., Smith, J.R., Bengio, Y.: Oracle performance for visual captioning. In: British Machine Vision Conference. (2016)

[312] Wu, Q., Shen, C., Liu, L., Dick, A., van den Hengel, A.: What value do explicit high level concepts have in vision to language problems? In: Proceed-

ings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 203–212

[313] Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: The 40th annual meeting of the Association for Computational Linguistics. (2002) 311–318

[314] Lavie, M.D.A.: Meteor universal: Language specific translation evaluation for any target language. In: The 52nd Annual Meeting of the Association for Computational Linguistics. (2014)

[315] Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: The 42nd Annual Meeting of the Association for Computational Linguistics Workshop. (2004)

[316] Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4566–4575

[317] Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollár, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)

[318] Liu, Y., Lew, M.S.: Learning relaxed deep supervision for better edge detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 231–240

# English Summary

It has long been the goal of computer vision researchers to develop an algorithm capable of understanding the visual information automatically and accurately. While this seems to be effortless to humans, there is no robust solution to date due to the well-known semantic gap between low-level features and the object or concept that is being modeled. In recent years, deep learning algorithms have been effective in closing this semantic gap due largely to the sophisticated visual representations they developed. This has resulted in major advances in diverse visual applications, such as image classification, object detection, image captioning and etc. The purpose of this thesis is to explore and design new deep learning algorithms for better visual understanding.

First, we present a comprehensive review of recent deep learning advances which targets general neural computing, computer vision and multimedia researchers who are interested in the state-of-the-art in deep learning in computer vision. Next, we establish our research on three visual applications: traditional image classification, hierarchical image classification and image captioning.

The traditional image classification task involves classifying an image into one pre-defined category, and has been widely studied in the computer vision community for decades. We proposed several new features, PPC and BoSP, to address this task. PPC is a straightforward scheme, which extracts and aggregates CNN features from different image regions, and utilizes PCA to reduce the feature dimension. BoSP regards the feature maps as surrogate parts, and proposes to assign the dense image regions to these surrogate parts by observing the activation

values. Both PPC and BoSP can be achieved without significantly increasing the computational cost.

Objects are often organised in a hierarchy. While the traditional image classification task only focuses on the leaf-level categories, we propose that providing an evolution of the image categories can better describe what the categories are. Accordingly, we introduce the hierarchical image classification task, which attempts to generate hierarchical coarse-to-fine labels rather than one leaf label, and develop the CNN-RNN framework to address this task. In this framework, the CNN is used to extract discriminative image features, and the RNN exploits the relationship between the hierarchical categories and generate sequential labels. In addition, we also investigate the effectiveness of utilizing this framework for the traditional image classification task and weakly supervised learning.

Another usage of the CNN-RNN framework is for image captioning, which is an important and challenging task in vision-to-language research. It aims to describe an image with meaningful and sensible sentence-level captions. We investigate the effects of different Convnets on image captioning, i.e. single-label Convnet, multi-label Convnet and multi-attribute Convnet. As these three Convnets focus on different visual contents in the image, we propose aggregating them together for a richer visual representation. Overall, we achieve competitive results with the state-of-the-art.

# Nederlandse Samenvatting

Al geruime tijd is het doel van 'Computer Vision'-onderzoekers een algoritme te ontwikkelen dat in staat is om automatisch en accuraat visuele informatie te begrijpen. Terwijl het lijkt dat mensen dit zonder enige inspanning kunnen is er tot op de dag van vandaag geen robuuste oplossing beschikbaar vanwege de bekende semantische kloof tussen het object of concept dat gemodelleerd wordt en haar low-level features. Recentelijk, zijn 'Deep Learning'-algoritmen effectief gebleken om deze semantische kloof te dichten voornamelijk als gevolg van de geavanceerde visuele representatie die zij hebben ontwikkeld. Dit heeft geresulteerd in substantiële verbeteringen in verschillende visuele toepassingen zoals beeld-classificatie, object-detectie, beeld-ondertiteling, etc. Het doel van deze thesis is om nieuwe 'Deep Learning'-algoritmen te onderzoeken en te ontwikkelen voor een beter begrip van visuele informatie.

Ten eerste geven we een uitgebreid en diepgaand overzicht van recente ontwikkelingen en vooruitgang op het gebied van 'Deep Learning'. Speciaal gericht op onderzoekers in de gebieden 'Neural Computing', 'Computer Vision' en 'Multimedia' die geïnteresseerd zijn in de state-of-the-art in 'Deep Learning in Computer Vision'. Vervolgens beschrijven we ons onderzoek op het gebied van drie visuele toepassingen: traditionele beeld-classificatie, hiërarchische beeld-classificatie en beeld-beschrijving (-'captioning').

De traditionele beeld-classificatie-taak bestaat uit het classificeren van een beeld in een van te voren gedefinieerde categorie. Dit probleem is binnen de 'Computer Vision'-gemeenschap gedurende verschillende decennia op brede schaal onderzocht. Om deze taak uit te voeren hebben we verschillende nieuwe beeld-

kenmerken voorgesteld: PPC en BoSP. PPC is een eenvoudig schema dat CNN-kenmerken extraheert en samenneemt vanuit verschillende regio's van het beeld en PCA toepast om de dimensie van de kenmerken te reduceren. BoSP beschouwt de afbeeldingen van de beeld-data op de kenmerken als surrogaat-delen en wijst aan de hand van de activatie-waarden de dichtbevolkte regio's van het beeld toe aan deze surrogaat-delen. Zowel PPC- als BoSP-kenmerken kunnen bepaald worden zonder een significante toename in de berekeningskosten.

Objecten zijn vaak georganiseerd in een hiërarchie. Terwijl de traditionele beeld-classificatie-taak zich enkel richt op de categorieën op het laagste niveau in de uiteinden van de hiërarchie, stellen wij dat de categorieën beter kunnen worden beschreven door de evolutie van de beeld-categorieën. Daarom introduceren we de hiërarchische beeld-classificatie-taak, welke tracht hiërarchische labels van grof naar fijn te genereren in plaats van een enkel label op het laagste niveau. Hiertoe ontwikkelen we een CNN-RNN raamwerk, waarbij de CNN wordt gebruikt om discriminatieve beeldkenmerken te extraheren en de RNN de relatie tussen de hiërarchische categorieën exploiteert om sequentiële labels te genereren. Bovendien onderzoeken we de effectiviteit van het gebruik van dit raamwerk voor de traditionele beeld-classificatie-taak en 'Weakly Supervised Learning'.

Een ander gebruik van het CNN-RNN raamwerk ligt bij de beschrijving van het beeld. Dit is een belangrijke en uitdagende taak in 'Vision-to-Language'-onderzoek. Hierbij is het doel om een beeld te beschrijven met betekenisvolle en zinnige beschrijvingen op het niveau van volledige zinnen. We onderzoeken de effecten van verschillende Convnets op de beeldbeschrijvingen, d.w.z. Single Label Convnet, Multi-Label Convnet en Multi-Attribute Convnet. Alle drie de Convnets richten zich op verschillende delen van de visuele inhoud van het beeld. We stellen voor om ze samen te nemen om zo een rijkere visuele representatie te verkrijgen. Over het algemeen bereiken we competitieve resultaten in vergelijking met de state-of-the-art.

# Acknowledgements

This thesis would not have been possible without the support and advice from so many people around me. I really cherish my studying time in media lab. It is an excellent environment of academic freedom, and inspires me a lot. Special thanks go to Prof. Dr. Songyang Lao. Thank you for leading me into my academic career, and giving me continuous support for so many years.

I would also like to express my appreciation to my colleagues in media lab, Song Wu, Yu Liu and Theodoros Georgiou. It is a great treasure for me to study with you. Song Wu gave me a lot of help when I first arrived to Leiden, enabling me to get used to the life here quickly. Yu Liu inspired me a lot for my research. I benefited a lot from your smart suggestions and academic spirit.

I am very thankful to my coworkers, Dr. Erwin M. Bakker, Dr. Ard Oerlemans, Maaike H.T. de Boer, Liang Bai and Li Liu. You contributed a lot to my research and taught me the art of scientific writing. It was really great to cooperate with you.

Many thanks go to my friends in the Netherlands: Feng Yao, Yunyan Xing, Zhiwei Yang, Minghao Li, Jian Yang, Wenbo Ma, Yuze Mu, Yaojin Peng, Zhenyu Xiao, Boyang Liu, Qiuyin Hu, Shengfa Miao, Di Liu, Yuanhao Guo, Fuyu Cai, Zhan Xiong, Junling He, Jun Lei, Kaifeng Yang, Channa Li, Hao Wang, Longmei Li, Peng Wang, Hongchan Shan, Xiaoqin Tang, Jun Li, Erqian Tang, Yazhou Yang, Xu Xie, Mengmeng Sun, Min He, Enrique Larios Vargas, Xiaoyu Chen, Guangsheng Du, Rui Zhang, Puning Liu, Yinlong Xiao, Guangchao Chen, Yuchuan Qiao, Feng Zhang, Yang Deng, Nan He, Xiang Li, Yun Tian, Feng Jiang, Lin Jiang, Yiwei Wang, Yingguang Li, Yang Yang, Yudan Tan, Ciqing Tong, Dan

## ACKNOWLEDGEMENTS

Hou, Jiao Shi, Yan Ren, Jiaqi Zhao, Zhiguo Zhou. Thank you for lighting my life abroad.

My great gratitude goes to my parents for their unconditional love and support. I am so lucky to be your son and my appreciation is beyond words. My final words go to my dearest wife, Jinxian, and my cutest daughter, Xinyao. Thank you for standing together with me through thick and thin. You are my biggest gain in the Netherlands. Let us live together to meet a better tomorrow.

# Curriculum Vitae

Yanming Guo was born in Hengshui, Hebei, China on May 16, 1989. In 2007, he started his study in the National University of Defense Technology in Changsha, Hunan, China, and received the B.S. degree and the M.S degree (under the supervision of Prof. Dr. Songyang Lao) in 2011 and 2013, respectively.

In October 2013, he got a scholarship from the China Scholarship Council and started his research at the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, the Netherlands, under the supervision of Prof. Dr. J.N. Kok and Dr. M.S. Lew. His research mainly focuses on developing deep learning algorithms for better visual understanding, including image classification, image retrieval, image captioning and cross-modal retrieval. He is a reviewer of the IEEE Transactions on Neural Networks and Learning Systems, Neurocomputing, IET Computer Vision, International Journal of Multimedia Information Retrieval and Journal of Ambient Intelligence and Smart Environments.