Multimed Tools Appl (2018) 77:10251–10271
https://doi.org/10.1007/s11042-017-5443-x

CNN-RNN: a large-scale hierarchical image classification framework

Yanming Guo[1] · **Yu Liu**[1] · **Erwin M. Bakker**[1] · **Yuanhao Guo**[1] · **Michael S. Lew**[1]

Received: 30 March 2017 / Revised: 6 November 2017 / Accepted: 20 November 2017 /
Published online: 12 December 2017

Abstract 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 paper, 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

✉ Yanming Guo
y.guo@liacs.leidenuniv.nl

Yu Liu
y.liu@liacs.leidenuniv.nl

Erwin M. Bakker
e.m.bakker@liacs.leidenuniv.nl

Yuanhao Guo
y.guo.3@liacs.leidenuniv.nl

Michael S. Lew
m.s.lew@liacs.leidenuniv.nl

[1] Leiden Institute of Advanced Computer Science, Leiden, Netherlands

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.

Keywords Convolutional neural network · Recurrent neural network · Hierarchical image classification · Wider-Resnet

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 [20, 47, 51].

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 [12, 35, 41], and have achieved quite accurate predictions on small datasets [4, 10].

As a consequence, the current research focus has moved to larger and more challenging datasets [3, 34], such as ImageNet [6]. 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 Fig. 1c, 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 paper 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:

(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 Fig. 1c.
(2) It can exploit the relationship between the coarse and fine categories, which, in turn, helps the traditional image classification task. For example, when we build the CNN-RNN framework with wrn-28-10 [53], 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

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

Fig. 1 The example images with the 'coarse → fine' label

experiments with three high-performance networks, i.e. CNN-7 [25], wrn-28-10 [53] 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 [7, 30], 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 [32].

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. Therefore, the second contribution of this paper is, we build a high performance network, i.e. wider-ResNet. In recent years, deep residual networks (ResNet) [12] have attracted great attention because of its leading performance in several image classification tasks and Zagoruyko et al. [53] 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. We intend to further enhance the performance and build a wider ResNet compared to [53]. Our implementation shows that, the wider-Resnet performs better than [53] 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 [6]. 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 Fig. 1a and 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 paper 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.

2 Related work

2.1 Usage of CNN-RNN framework

In recent years, deep learning methods have attracted significant attention [11] and have achieved revolutionary successes in various applications [12, 24]. 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 [8, 23, 44, 48, 52, 54] have attempted to combine them together, and have 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. [54] 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 [46]. 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. [48] 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, [8, 45, 52] employed CNN-RNN to address the image captioning task, and [50] utilized CNN-RNN to rank the tag list based on the visual importance.

In this paper, 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.

2.2 Hierarchical models for image classification

Hierarchical models have been used extensively for image classification. For example, Salakhutdinov et al. [33] presented a hierarchical classification model to share features between categories, and boosted the classification performance for objects with few training examples. Yan et al. [49] presented a hierarchical deep CNN (HD-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. [29] 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. [7] introduced HEX graphs to capture the hierarchical and exclusive relationship between categories. Ristin et al. [30] 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 paper 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 [7, 30] that can only process the pre-computed image features, our proposed CNN-RNN framework can be trained end-to-end.

3 Our proposed scheme

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.

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 a common practice to fulfill this specific task. In this paper, 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 (Fig. 2: Strategy 1 & 2), or in a parallel pattern (Fig. 2: Strategy 3).

During the training phase, we utilize the softmax loss function to jointly optimize the coarse and fine label predictions, as defined in (1).

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

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.

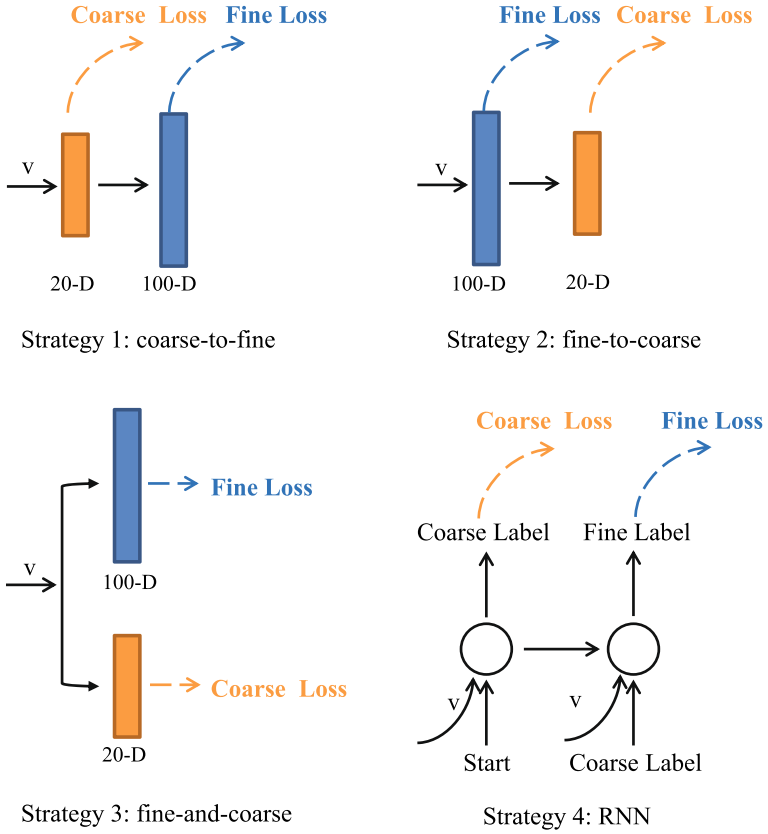

Fig. 2 The illustration of the four strategies which can jointly train and generate the coarse and fine labels

3.2 CNN-RNN generator

A CNN-RNN generator determines hierarchical predictions using an architecture where the last layer of a CNN is replaced by an RNN (Fig. 2: Strategy 4).

RNN [9] is a class of artificial neural networks where connections between units form a directed cycle, as shown in Fig. 3. It can effectively model the dynamic temporal behavior of sequences with arbitrary lengths. However, RNN suffers from the vanishing and exploding gradient problem since the gradients need to propagate down through many layers of the recurrent network. Therefore, it is difficult to model the long-term dynamics. In contrast, Long-Short Term Memory (LSTM) [14] provides a solution by incorporating a memory cell to encode knowledge at each time step. Specifically, the behavior of the cell is controlled by three gates: an input gate, a forget gate and an output gate. These gates are used to control how much it should read its input (input gate i), whether to forget the current cell value (forget gate f) and whether to output the new cell value (output gate o). These gates help the input signal to prorogate through the recurrent hidden states without affecting the output, therefore, LSTM can deal well with exploding and vanishing gradients, and effectively model long-term temporal dynamics that RNN is not capable of learning.

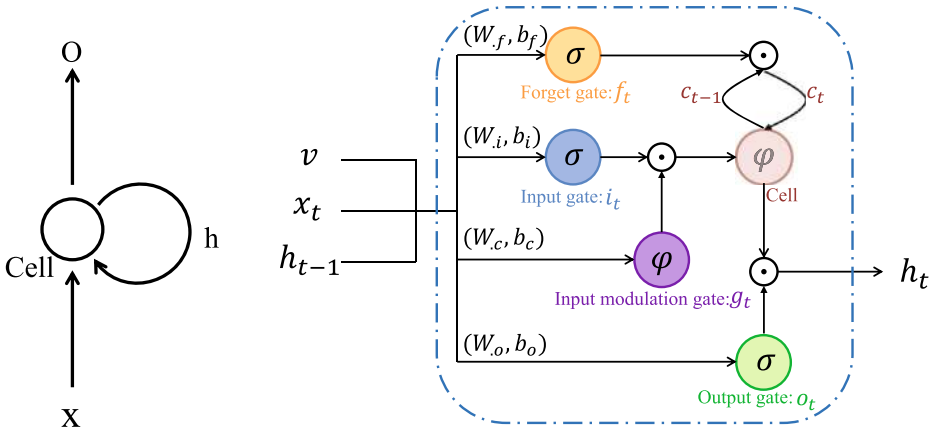

Fig. 3 The pipeline of RNN(left) and LSTM(right)

In this paper, 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:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{vi}v + b_i) \tag{2}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{vf}v + b_f) \tag{3}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{vo}v + b_o) \tag{4}$$

$$g_t = \varphi(W_{xc}x_t + W_{hc}h_{t-1} + W_{vc}v + b_c) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{6}$$

$$h_t = o_t \odot \varphi(c_t) \tag{7}$$

Where $\odot$ represents the product operation, σ is the sigmoid function ($\sigma(x) = (1 + e^{-x})^{-1}$), and φ is the hyperbolic tangent function ($\varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$). 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 Fig. 4.

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 (8).

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T}\sum_{j=1}^{C+F} 1\left\{x_t^i = j\right\} \log p_j\right) \tag{8}$$

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,

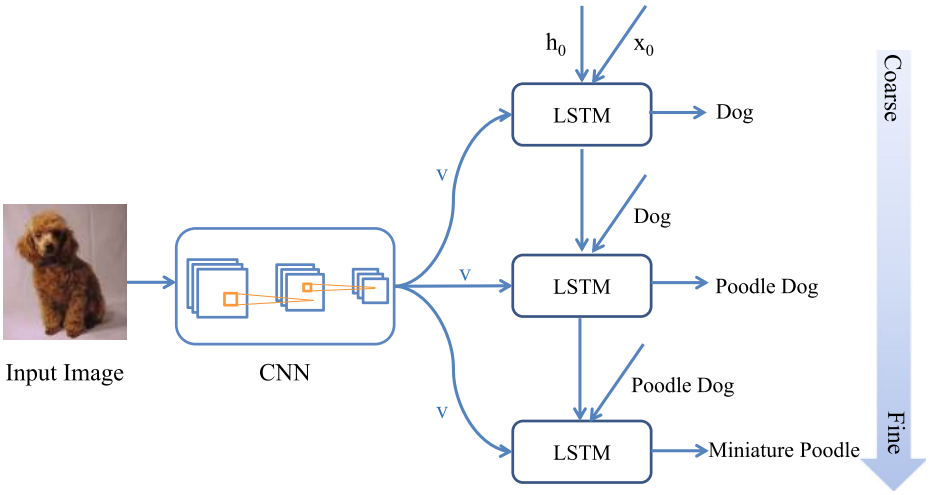

Fig. 4 The pipeline of the CNN-RNN framework

i.e. $W_t = argmax_{W_{t-1}} p(W_{t-1}|I)$, and then utilize the predicted label as the input for the next timestep.

Furthermore, we visualize the LSTM activations of different timesteps using the t-SNE technique [43] on CIFAR-100 [18], as shown in Fig. 5. The two activations are intended for the coarse and fine predictions, respectively. We can see that, the activation of the second timestep is more discriminative than that of the first 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.

4 Experiments

We perform our experiments on three well-known datasets: CIFAR-100 [18], ImageNet 2012 [32] and a subset of ImageNet 2010 [30]. These three datasets have provided hierarchical image labels. The characteristics of the three datasets are summarized in Table 1.

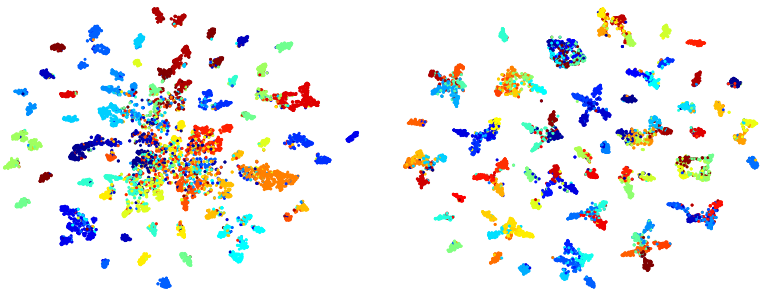

Fig. 5 The visualization of the LSTM activations of different timesteps. The left one is the feature intended for the coarse label classification, while the right one is used for the fine label classification

Table 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 [16] 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.

4.1 Hierarchical predictions

We evaluate the hierarchical predictions on two widely-used datasets: CIFAR-100 [18] and ImageNet 2012 [32].

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 32×32 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 2.

All of the evaluations in this part are conducted based on the CNN model proposed in [25], because of its high training efficiency and decent performance on CIFAR-100. The

Table 2 The comparison of the accuracy for the coarse categories and fine categories. Best results are in bold face

	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%**

'+' indicates a standard data augmentation (translation/mirroring)

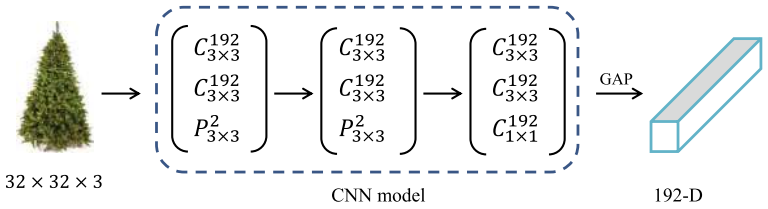

Fig. 6 The CNN baseline proposed in [25]

CNN structure is shown in Fig. 6. We employ exactly the same experimental configuration as used in [25].

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 [25], wrn-28-10 [53] and our proposed wider-Resnet.

For wrn-28-10, we adopt the version with dropout [39], and train the network with larger mini-batch size (i.e. 200), and more iterations (a total of 7×10^4 iterations, and the learning rate dropped at 2×10^4, 4×10^4, 6×10^4 iterations). Other experimental configuration follows [53].

The structure of our proposed wider-Resnet is shown in Table 3. We adopt the pre-activation residual block as in [13], and train the models for a total of 7×10^4 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 4×10^4 and 6×10^4 iterations.

The results on these three datasets are shown in Table 4. 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 [53] cannot reproduce the original published results, possibly as a result of the differences in the platforms (Torch v.s. Caffe), or the differences

Table 3 The framework of our proposed wider-Resnet

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

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.

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 [53] 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

Table 4 The comparison of the accuracy for the coarse categories and fine categories. For each network, CNN-RNN could get better results and their results are bolded

		C100+	
		Coarse	Fine
CNN-7 [25]	coarse-specific	82.09%	–
	fine-specific	–	72.03%
	CNN-RNN	**83.21%**	**72.26%**
wrn-28-10 [53]	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%**

'+' indicates standard data augmentation (translation/mirroring)

Table 5 The test error of different methods on the CIFAR-100 dataset with standard data augmentation (translation/mirroring). Best results are in bold face

Method	C100+
FitNet [31]	35.04%
DSN [21]	34.57%
All-CNN [38]	33.71%
Highway Network [40]	32.39%
APL [1]	30.83%
SReLU [17]	29.91%
BayesNet [37]	27.4%
Fitnet4-LSUV [28]	27.66%
ELU [5]	24.28%
MBA [22]	24.1%
ResNet-110 [12] (according to [15])	27.22%
ResNet-110 (Stochastic Depth) [15]	24.58%
ResNet-164 (Pre-activation) [13]	24.33%
ResNet-1001 (Pre-activation) [13]	22.71%
18-layer + wide RiR [42]	22.90%
FractalNet-20 [19]	23.30%
FractalNet-40 [19]	22.49%
SwapOut V2 [36] (width×4)	22.72%
wrn-28-10 [53] (our reproduced)	25.45%
wrn-28-10-RNN	23.77%
wider-Resnet	22.03%
wider-Resnet-RNN	**20.86%**

are still seeking to build the CNN-RNN framework on top of future state-of-the-art architectures to boost the classification performance.

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 [32].

ImageNet is an image dataset organized according to the WordNet hierarchy [27]. 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 [12] 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 224×224 pixels and extract the last pooling features utilizing ResNet-152, and then send the features into LSTM for modeling the category dependency.

Figure 7 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

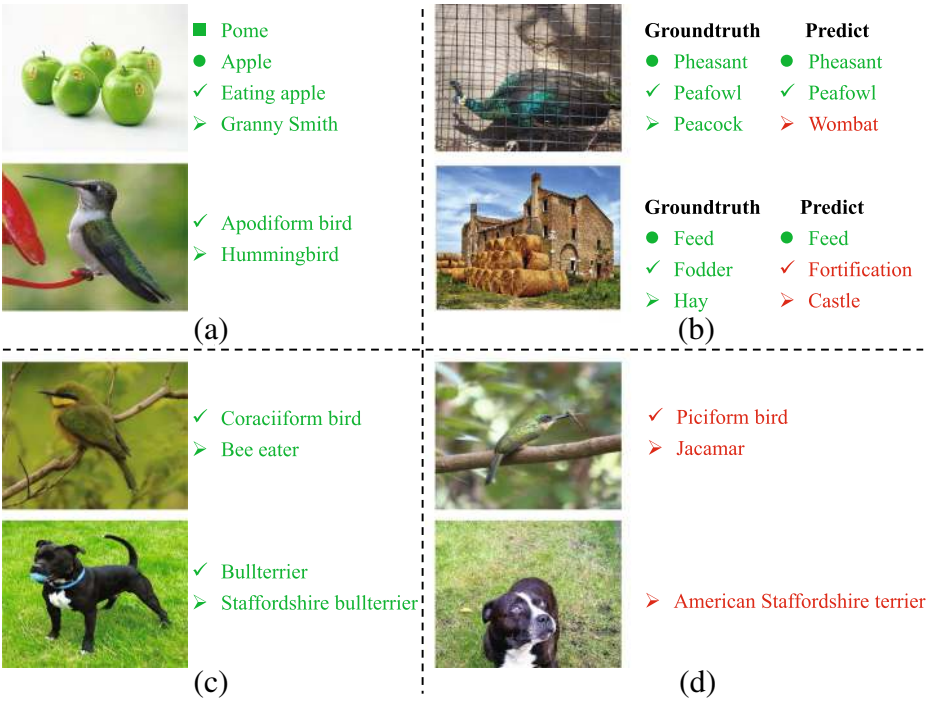

Fig. 7 The hierarchical predictions of some example images. **a** and **c** show some positive examples. **b** shows the examples with partly wrong predictions, e.g. correct coarse labels & wrong fine labels. **d** shows examples in the same category as (**c**), but which have totally wrong predictions

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 Fig. 7a, 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 Fig. 7b.

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 [30].

We follow the experiment setup of [30], and conduct our experiment on a subset of ImageNet 2010. This dataset particularly selected the classes from ImageNet 2010 that have

Table 6 Accuracy for classifying fine labels using the ImageNet 2010 subset described in [30]

	Training Set	Accuracy
NCM [26]	S	66.02%
Multiclass SVM [2]	S	71.67%
RNCMF [30]	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%

RNN: train the RNN with extracted image features from VGG-Net [35]; CNN: finetune the VGG-Net [35] on the ImageNet 2010 subset; CNN-RNN: jointly train the VGG-Net [35] 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

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. More details about the dataset can be found in (http://www.vision.ee.ethz.ch/datasets_extra/mristin/ristin_et_al_cvpr15_data.zip).

All of the image features are extracted from the VGG-Net [35], as was done for the preliminary experiments in [30].

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 6.

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 [30], 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%.

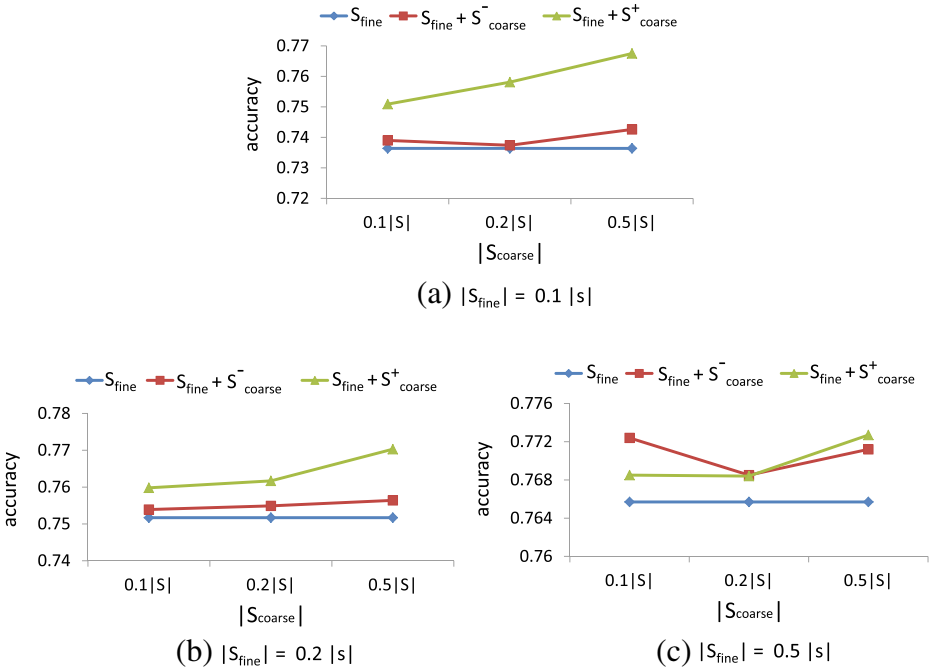

Fig. 8 The classification performance with different training/test set

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_{coarse} and S_{fine}. S_{coarse} has only the coarse labels, while S_{fine} has both coarse and fine labels. We vary $|S_{fine}| \in \{0.1|S|, 0.2|S|, 0.5|S|\}$, and for each S_{fine}, we further vary $|S_{coarse}| \in \{0.1|S|, 0.2|S|, 0.5|S|\}$.

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

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

The results are shown in Fig. 8.

In general, $S_{fine} + S^{-}_{coarse}$ performs better than S_{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.

Since the fine labels of S_{coarse} are predicted by the RNN trained on S_{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 $|S_{fine}|$ is small. As we can see in Fig. 8a, when $|S_{fine}| = 0.1|S|$, the classification hardly benefited from using S_{coarse} when compared to the RNN trained solely on S_{fine}.

Table 7 Accuracy in classifying fine categories for the test set. We set the amount of coarse-labeled data to $|S_{coarse}| = 0.5|S|$. Best results are in bold face

	$\|S_{fine}\|$		
	$0.1\|S\|$	$0.2\|S\|$	$0.5\|S\|$
RNCMF [30]	68.49%	70.49%	73.07%
NN-H-RNCMF [30]	69.95%	71.41%	73.43%
RNN	**74.26%**	**75.64%**	**77.12%**

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

We further compare our method with the NN-H-RNCMF [30], which also attempted to improve the classification by exploiting the hierarchy. We set the amount of coarse-labeled data to $|S_{coarse}| = 0.5|S|$, and vary $|S_{fine}| \in \{0.1|S|, 0.2|S|, 0.5|S|\}$, and the results are shown in Table 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.

5 Conclusion

In this paper, 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.

References

1. Agostinelli F, Hoffman M, Sadowski P, Baldi P (2015) Learning activation functions to improve deep neural networks. In: International conference on learning representations workshops
2. Akata Z, Perronnin F, Harchaoui Z, Schmid C (2014) Good practice in large-scale learning for image classification. IEEE Trans Pattern Anal Mach Intell 36(3):507–520
3. Cao L, Gao L, Song J, Shen F, Wang Y (2017) Multiple hierarchical deep hashing for large scale image retrieval. Multimed Tools Appl 1–14
4. Cimpoi M, Maji S, Kokkinos I, Vedaldi A (2016) Deep filter banks for texture recognition, description, and segmentation. Int J Comput Vis 118(1):65–94

5. Clevert Dj, Unterthiner T, Hochreiter S (2016) Fast and accurate deep network learning by exponential linear units (elus). In: International conference on learning representations
6. Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F (2009) Imagenet A large-scale hierarchical image database. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 248–255
7. Deng J, Ding N, Jia Y, Frome A, Murphy K, Bengio S, Li Y, Neven H, Adam H (2014) Large-scale object classification using label relation graphs. In: European conference on computer vision, pp 48–64
8. Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2625–2634
9. Elman JL (1990) Finding structure in time. Cogn Sci 14(2):179–211
10. Guo Y, Lew MS (2016) Bag of surrogate parts: one inherent feature of deep cnns. In: British machine vision conference
11. Guo Y, Liu Y, Oerlemans A, Lao S, Song W, Lew MS (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 770–778
13. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: European conference on computer vision, pp 630–645
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
15. Huang G, Sun Y, Liu Z, Sedra D, Weinberger KQ (2016) Deep networks with stochastic depth. In: European conference on computer vision, pp 646–661
16. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the ACM international conference on multimedia, pp 675–678
17. Jin X, Chunyan X, Feng J, Wei Y, Xiong J, Yan S (2016) Deep learning with s-shaped rectified linear activation units. In: AAAI, pp 1737–1743
18. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images
19. Larsson G, Maire M, Shakhnarovich G (2017) Fractalnet: ultra-deep neural networks without residuals. In: International conference on learning representations
20. Lazebnik S, Schmid C, Ponce J (2006) Beyond bags of features Spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2169–2178
21. Lee C-Y, Xie S, Gallagher P, Zhang Z, Zhuowen T (2015) Deeply-supervised nets. In: Artificial intelligence and statistics, pp 562–570
22. Li H, Ouyang W, Wang X (2016) Multi-bias non-linear activation in deep neural networks. In: International conference on machine learning, pp 221–229
23. Liang M, Xiaolin H (2015) Recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3367–3375
24. Liu Y, Guo Y, Song W, Lew MS (2015) Deepindex for accurate and efficient image retrieval. In: Proceedings of the ACM on international conference on multimedia retrieval, pp 43–50
25. Liu Y, Guo Y, Lew MS (2017) On the exploration of convolutional fusion networks for visual recognition. In: International conference on multimedia modeling, pp 277–289
26. Mensink T, Verbeek J, Perronnin F, Csurka G (2013) Distance-based image classification: feneralizing to new classes at near-zero cost, vol 35
27. Miller GA (1995) Wordnet: a lexical database for english. Commun ACM 38(11):39–41
28. Mishkin D, Matas J (2016) All you need is a good init. In: International conference on learning representations
29. Murdock C, Li Z, Zhou H, Duerig T (2016) Blockout: dynamic model selection for hierarchical deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2583–2591
30. Ristin M, Gall J, Guillaumin M, Gool LV (2015) 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, pp 231–239
31. Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y (2015) Fitnets: hints for thin deep nets. In: International conference on learning representations
32. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
33. Salakhutdinov R, Torralba A, Tenenbaum J (2011) Learning to share visual appearance for multiclass object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1481–1488

34. Shirahama K, Grzegorzek M (2016) Towards large-scale multimedia retrieval enriched by knowledge about human interpretation. Multimed Tools Appl 75(1):297–331
35. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations
36. Singh S, Hoiem D, Forsyth D (2016) Swapout: learning an ensemble of deep architectures. In: Advances in neural information processing systems, pp 28–36
37. Snoek J, Rippel O, Swersky K, Kiros R, Satish N, Sundaram N, Patwary M, Mr P, Adams R (2015) Scalable bayesian optimization using deep neural networks. In: International conference on machine learning, pp 2171–2180
38. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M (2015) Striving for simplicity: the all convolutional net. In: International conference on learning representations workshops
39. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
40. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks. In: International conference on learning representations workshops
41. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 1–9
42. Targ S, Almeida D, Lyman K (2016) Resnet in resnet: generalizing residual architectures. In: International conference on learning representations workshops
43. van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9:2579–2605
44. Vinyals O, Toshev A, Bengio S, Erhan D (2015) Show and tell: a neural image caption generator. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 3156–3164
45. Vinyals O, Toshev A, Bengio S, Erhan D (2017) Show and tell: lessons learned from the 2015 mscoco image captioning challenge. IEEE Trans Pattern Anal Mach Intell 39(4):652–663
46. Visin F, Kastner K, Cho K, Matteucci M, Courville A, Bengio Y (2015) Renet: a recurrent neural network based alternative to convolutional networks. arXiv:1505.00393
47. Wang J, Yang J, Kai Y, Lv F, Huang T, Gong Y (2010) Locality-constrained linear coding for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3360–3367
48. Wang J, Yi Y, Mao J, Huang Z, Huang C, Xu W (2016) Cnn-rnn: a unified framework for multi-label image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2285–2294
49. Yan Z, Zhang H, Piramuthu R, Jagadeesh V, DeCoste D, Di W, Yizhou Y (2015) Hd-cnn: hierarchical deep convolutional neural networks for large scale visual recognition. In: Proceedings of the IEEE international conference on computer vision, pp 2740–2748
50. Yan G, Wang Y, Liao Z (2016) Lstm for image annotation with relative visual importance. In: British machine vision conference
51. Yang J, Kai Y, Gong Y, Huang T (2009) Linear spatial pyramid matching using sparse coding for image classification. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 1794–1801
52. You Q, Jin H, Wang Z, Fang C, Luo J (2016) Image captioning with semantic attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4651–4659
53. Zagoruyko S, Komodakis N (2016) Wide residual networks. In: British machine vision conference
54. Zuo Z, Shuai B, Wang G, Liu X, Wang X, Wang B, Chen Y (2015) Convolutional recurrent neural networks: learning spatial dependencies for image representation. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 18–26

Yanming Guo received the B.S. degree in information system engineering, the M.S degree in operational research from the National University of Defense Technology, Changsha, China, in 2011 and 2013, respectively. He is currently a Ph.D. in Leiden Institute of Advanced Computer Science (LIACS), Leiden University. His current research interests include image classification, image captioning, object detection and image retrieval.

Yu Liu received the B.S. degree and M.S degree from School of Software Technology, Dalian University of Technology, Dalian, China, in 2011 and 2014, respectively. He is currently a Ph.D. in Leiden Institute of Advanced Computer Science (LIACS), Leiden University. His current research interests include semantic segmentation and image retrieval.

Erwin M. Bakker is co-director of the LIACS Media Lab at Leiden University. He has published widely in the fields of image retrieval, audio analysis and retrieval and bioinformatics. He was closely involved with the start of the International Conference on Image and Video Retrieval (CIVR) serving on the organizing committee in 2003 and 2005. Moreover, he regularly serves as a program committee member or organizing committee member for scientific multimedia and human-computer interaction conferences and workshops.

Yuanhao Guo received his M.S. degree in image processing and pattern recognition from Shandong University, China, in 2009. He is currently a Ph.D candidate in Leiden Institute of Advanced Computer Science at Leiden University and working on the light microscope imaging and 3D reconstruction. He is also interested in the applications of machine learning and computer vision. More information can be seen from http://liacs.leidenuniv.nl/~guoy3/.

Michael S. Lew is co-head of the Imagery and Media Research Cluster at LIACS and director of the LIACS Media Lab. He received his doctorate from University of Illinois at Urbana-Champaign and then became a postdoctoral researcher at Leiden University. One year later he became the first Leiden University Fellow which was a pilot program for tenure track professors. In 2003, he became a tenured associate professor at Leiden University and was invited to serve as a chair full professor in computer science at Tsinghua University (the MIT of China). He has published over 100 peer reviewed papers with three best paper citations in the areas of computer vision, content-based retrieval, and machine learning. Currently (September 2014), he has the most cited paper in the history of the ACM Transactions on Multimedia. In addition, he has the most cited paper from the ACM International Conference on Multimedia Information Retrieval (MIR) 2008 and also from ACM MIR 2010. He has served on the organizing committees for over a dozen ACM and IEEE conferences. He served as the founding the chair of the ACM ICMR steering committee and had served as chair for both the ACM MIR and ACM CIVR steering committees. In addition he is the Editor-in-Chief of the International Journal of Multimedia Information Retrieval (Springer) and a member of the ACM SIGMM Executive Board which is the highest and most influential committee of the SIGMM.