Knowledge Guided Disambiguation for Large-Scale Scene Classification with Multi-Resolution CNNs

Limin Wang, Sheng Guo, Weilin Huang, *Member, IEEE,* Yuanjun Xiong, and Yu Qiao, *Senior Member, IEEE*

***Abstract*—Thanks to the available large-scale scene datasets such as Places and Places2, Convolutional Neural Networks (CNNs) have made remarkable progress on the problem of scene recognition. However, scene categories are often defined according its functions and there exist large intra-class variations in a single scene category. Meanwhile, as the number of scene classes is increasing, some classes tend to overlap with others and label ambiguity is becoming a problem. This paper focuses on large-scale scene recognition and makes two major contributions to tackle these issues. First, we propose a multi-resolution CNN architecture to capture visual content and structure at different scales. Our proposed multi-resolution CNNs are composed of coarse resolution CNNs and fine resolution CNNs, whose performance is complementary to each other. Second, we design two knowledge guided disambiguation techniques to deal with the problem of label ambiguity. In the first scenario, we exploit the knowledge from confusion matrix at validation data to merge similar classes into a super category, while in the second scenario, we utilize the knowledge of extra networks to produce a soft label for each image. Both the information of super category and soft labels are exploited to train CNNs on the Places2 datasets. We conduct experiments on three large-scale image classification datasets (ImangeNet, Places, Places2) to demonstrate the effectiveness of our proposed approach. In addition, our method takes part in two major scene recognition challenges, and we achieve the 2^{nd} place at the Places2 challenge 2015 and 1^{st} place at the LSUN challenge 2016. Finally, we transfer the learned representations to the datasets of MIT Indoor67 and SUN397, which yields the state-of-the-art performance (86.7% and 72.0%) on both datasets.**

***Index Terms*—Scene recognition, Convolutional neural networks, multi-resolutions, disambiguation.**

I. Introduction

Scene recognition [1], [2] is a fundamental and important problem in computer vision and has received a large number of research attention in the past few years [3], [4], [5], [6], [7], [8], [9]. Scene recognition not only provides semantic information of global structure [10], but also yields context to assist other vision tasks like object detection [11], [12], event recognition [13], [14], and action recognition [15], [16]. In general, it is assumed that scene is composed of specific objects arranged in a certain layout. Cognitive evidence has implied that human vision system is highly sensitive to the global structure and special regions of an image, while puts little attention to the local objects and features outside of the attentional regions. Therefore, compared with object, the concept of scene is more subjective, and there may not exist consensus on how to determine an environment category, which poses more challenges for developing effective and robust scene recognition algorithms in computer vision research.

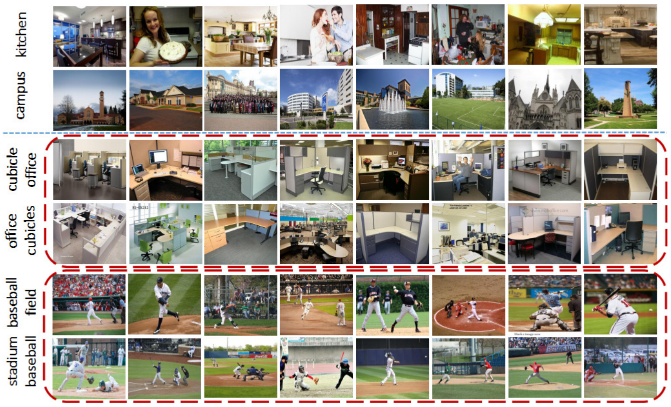

Fig. 1. Image examples from the Places2 dataset. In top row, we show images from two separate scene classes (i.e. kitchen and campus). We notice that large intra-class variations are contained in these images. In bottom row, we give two pairs of scene categories (i.e. (cubicle office, office cubicles), (baseball field, stadium baseball)). We see that images from these scene classes are easily confused with those of the other class of the pair.

Recently, large-scale scene datasets (e.g. Places [1] and Places2 [17]) have been introduced to advance the research of scene understanding which allows to train powerful convolutional neural networks (CNNs) [18] for scene classification. These large-scale datasets consist of a rich scene taxonomy, which includes rich categories to cover the diverse visual environments of our daily experience. After having these scene categories, scene keywords are sent to image search engines (e.g. Google Images, Bing Images and Flicker) and millions of images are downloaded, which are further sent to Amazon Mechanical Turk for manual annotation. However, as the number of classes is rapidly growing, these visual categories start to overlap with each other and there exists label ambiguity among these scene classes. As shown in Figure 1, *cubicle office* and *office cubicles* include confused images which may be easily identified as the other category, so do *baseball field* and *stadium baseball*. Partially due to this reason, the human top-1 error rate is still relatively high on the SUN397 dataset [2] (around 30%).

L. Wang was with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, and is with Computer Vision Lab, ETH Zurich, Switzerland, (e-mail: 07wanglimin@gmail.com)

S. Guo is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, (e-mail: sheng.guo@siat.ac.cn)

W. Huang is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, and is also with Visual Geometry Group, Oxford University, UK, (e-mail: wl.huang@siat.ac.cn)

Y. Xiong is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, (e-mail: yjxiong@link.cuhk.edu.hk)

Y. Qiao is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, (e-mail: yu.qiao@siat.ac.cn)

Due to the inherent uncertainty of scene concepts and the increasing overlap among different scene categories, it is challenging to conduct scene recognition on the large-scale datasets (with hundreds of classes and millions of images). Specifically, the current large-scale scene datasets present two major challenges for scene classification, namely *visual inconsistence* and *label ambiguity*.

- For **visual inconsistence**, we refer to the fact that there exist large variations among the images from the same scene category. As there is no precise definition for scene, people label natural images according to their own experience which leads to large diversity on scene datasets. As shown in Figure 1, for instance, the category of *kitchen* contains very diverse images, ranging from the whole room with many cooking wares to a single people with food.
- For **label ambiguity**, we argue that some scene categories share similar visual appearance and could be easily confused with others. As the number of scene classes increases, the inter-category overlaps can become large. For example, as shown in Figure 1, the scene category of *baseball field* are very similar to the class of *stadium baseball*, and they both contain the representative objects such as track and people.

These challenges motivate us to make two major contributions for large-scale scene recognition: (1) *we propose a multi-resolution convolutional architecture to capture multi-level visual cues of different scales*; (2) *We introduce knowledge guided strategies to disambiguate similar scene categories*. **First**, to deal with the problem of visual inconsistence (i.e. large intra-class variations), we come up with a multi-resolution CNN framework, where CNNs at coarse resolution are able to capture the appearance of larger objects, while CNNs at fine resolution are capable of describing detailed local information of smaller objects. Intuitively, multi-resolution CNNs combine complementary visual cues at different scales and are good at tackling the issue of large intra-class variations. **Second**, for the challenge of label ambiguity (i.e. small inter-class variations), we propose to reorganize the semantic scene space to release the difficulty of training CNNs by exploiting extra knowledge. In particular, we design two methods with the assistance from confusion matrix on the validation dataset and publicly available CNN models, respectively. In the first method, we investigate the correlation of different classes and progressively merge similar categories into a super category. In the second method, we use the outputs of extra CNN models as new labels. These two methods essentially utilize extra knowledge to produce new labels for training images, and these new supervision signal is able to make the training of CNN easier or act as the regularizers to guide the CNN optimization.

To verify the effectiveness of our proposed method, we choose the successful BN-Inception architecture [19] as our network structure, and demonstrate the effectiveness of multi-resolution CNNs and knowledge guided disambiguation strategies on a few benchmarks. More specifically, we first conduct experiments on three large-scale image recognition datasets, including ImageNet [20], Places [1], and Places2 [17], where our method obtains highly competitive performance. Then, we apply our proposed framework on two important scene recognition challenges, namely the Places2 challenge 2015 (held with ImangeNet large scale visual recognition challenge [21]) and the large-scale scene understanding (LSUN) challenge 2016. Our team secures the 2^{nd} place at the Places2 challenge 2015 and 1^{st} place at the LSUN challenge 2016. Furthermore, we examine the generalization ability of our learned models and test them on the datasets of MIT Indoor67 [22] and SUN397 [2]. We obtain the current state-of-the-art performance on these two datasets. Finally, we show some failure cases produced by our method to highlight the existing challenges for scene recognition and possible research directions in the future.

The rest of this paper is organized as follows. In Section II, we review related works to our method from aspects of scene recognition, deep networks for image recognition, and knowledge transferring. Section III introduces the architecture of multi-resolution convolutional neural networks. In Section IV, we develop two types of knowledge guided disambiguation strategies to improve the performance of scene recognition. We report our experimental results and analyze different aspects of our method in Section VI. Finally, we conclude our method in Section VI.

II. Related Works

In this section, we briefly review the previous works that are related to ours, and clarify the difference between our work and the others. Specifically, we review previous works from three aspects: (1) scene recognition, (2) deep networks for image recognition, and (3) knowledge transferring.

Scene recognition. The problem of scene recognition has been extensively studied by previous works from different angles. For example, Lazebnik *et al.* [23] proposed spatial pyramid matching (SPM) to incorporate spatial layout into bag-of-word (BoW) representation for scene recognition. Partizi *et al.* [24] designed a reconfigurable version of SPM, which associated different BoW representations with different image regions. The standard deformable part model (DPM) [12] was extended to scene recognition by Pandey *et al.* [25]. Quattoni *et al.* [22] studied the problem of indoor scene recognition by modeling the spatial layout of scene components. Mid-level discriminative patches or parts were discovered and identified for scene recognition in [26], [27]. Recently, deep convolutional networks have been exploited for scene classification by Zhou *et al.* [1], where they introduced a large-scale places dataset and advanced the state of the art of scene recognition by a large margin. After this, they introduced another more challenging dataset [17] with more categories and images, called as Places2.

Our work differs from these previous methods mainly from two aspects: (1) We test our proposed method on a much larger dataset and processing dataset of such scale is challenging; (2) We design a multi-resolution architecture and propose a knowledge guided disambiguation strategy to improve the performance of scene recognition.

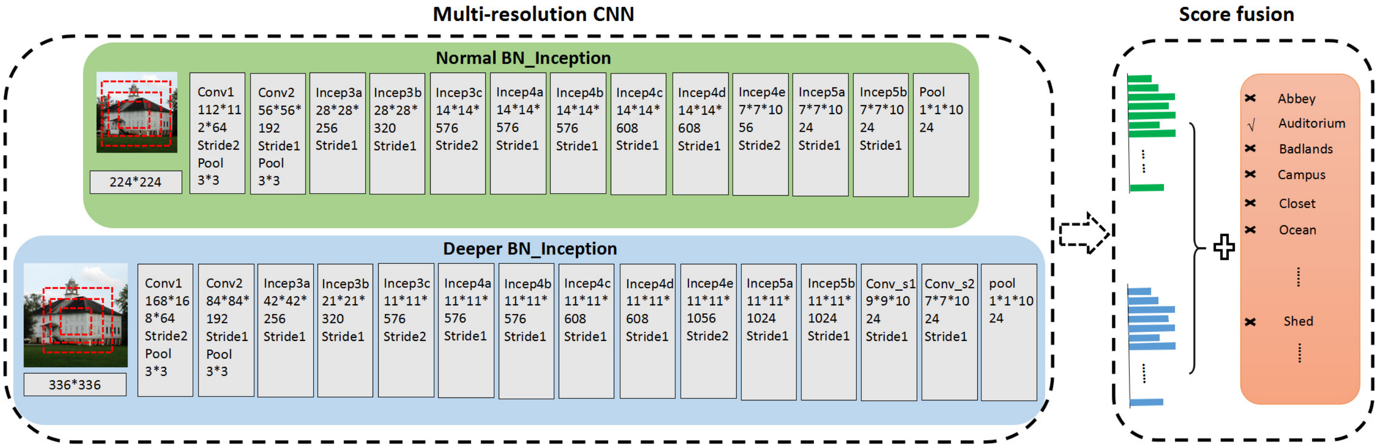
Multi-resolution CNN
Score fusion
Normal BN_Inception
224*224
Conv1 112*112*64 Stride2 Pool 3*3
Conv2 56*56*192 Stride1 Pool 3*3
Incep3a 28*28*256 Stride1
Incep3b 28*28*320 Stride1
Incep3c 14*14*576 Stride2
Incep4a 14*14*576 Stride1
Incep4b 14*14*576 Stride1
Incep4c 14*14*608 Stride1
Incep4d 14*14*608 Stride1
Incep4e 7*7*1056 Stride2
Incep5a 7*7*1024 Stride1
Incep5b 7*7*1024 Stride1
Pool 1*1*1024
Deeper BN_Inception
336*336
Conv1 168*168*64 Stride2 Pool 3*3
Conv2 84*84*192 Stride1 Pool 3*3
Incep3a 42*42*256 Stride1
Incep3b 21*21*320 Stride1
Incep3c 11*11*576 Stride2
Incep4a 11*11*576 Stride1
Incep4b 11*11*576 Stride1
Incep4c 11*11*608 Stride1
Incep4d 11*11*608 Stride1
Incep4e 11*11*1056 Stride2
Incep5a 11*11*1024 Stride1
Incep5b 11*11*1024 Stride1
Conv_s1 9*9*1024 Stride1
Conv_s2 7*7*1024 Stride1
pool 1*1*1024
✗ Abbey
√ Auditorium
✗ Badlands
✗ Campus
✗ Closet
✗ Ocean
✗ Shed

Fig. 2. **Multi-resolution CNN**: we propose a multi-resolution architecture, which is composed of coarse resolution CNN (normal bn-inception) and fine resolution CNN (deeper bn-inception). Coarse resolution CNNs capture visual structure at a large scale, while fine resolution CNNs describe visual pattern at a relatively smaller scale. The receptive fields (red boxes) of two CNNs correspond to regions of different scales and so their prediction scores are complementary, which are fused by taking arithmetic mean.

Deep networks for image recognition. Since the remarkable progress made by AlexNet [28] on ILSVRC 2012, great efforts have been devoted to the problem of image recognition with deep learning techniques [29], [30], [31], [32], [19], [33], [9], [34], [35]. A majority of these works focused on designing deeper network architectures, such as VGGNet [31], Inception Network [32], [34], and ResNet [35], finally containing hundreds of layers. Meanwhile, several regularization techniques and data augmentations have been designed to reduce the overfitting effective of network training, such as dropout [28], smaller convolutional kernel size [29], [31], and multi-scale cropping [31]. In addition, several optimization techniques have been also proposed to reduce the difficulty of training networks and improve recognition performance, such as Batch Normalization (BN) [19] and Relay Back Propagation [9].

These works focused on the general aspect of applying deep networks for image classification, in particular for object recognition, without considering the specifics of scene recognition problem. Complementary to these works, we conduct a dedicated study on the difficulty of scene recognition and accordingly come up with two new solutions to address the issues existed in scene recognition. We propose a multi-resolution architecture to capture visual information from different scales and hopefully to deal with the visual inconsistence problem. In addition, we design a knowledge guided disambiguation mechanism to handle the issue of label ambiguity, which is a another major challenge for scene recognition.

Knowledge transferring. Knowledge distillation or knowledge transferring from CNN models is becoming an important topic recently [36], [37], [38], [39], [40]. The basic idea of using network outputs as supervision signal to train other models was invented by Bucila *et al.* [41]. Recently, Hinton *et al.* [36] adopted this technique to compress model ensembles into a smaller model for fast deployment. Romero *et al.* [37] utilized this technique to help train deeper network in multiple stage. Tzeng *et al.* [39] explored this method in the problem of domain adaption for object recognition. Gupta *et al.* [38] proposed to distill knowledge across different modalities and used RGB CNN models to guide the training of CNNs for depth maps or optical flow field. Zhang *et al.* [40] developed a knowledge transfer technique to exploit soft codes of flow CNNs to assist the training of motion vector CNNs, with a goal of real-time action recognition from videos.

Our utilization of soft codes as supervision signal differs from these methods mainly from two points: (1) We conduct knowledge transfer crossing different visual tasks (e.g. object recognition vs. scene recognition), while previous methods all focus on the same task; (2) We exploit these soft codes to help circumvent the label ambiguity problem existed in large-scale scene recognition.

III. Multi-Resolution Convolutional Neural Networks

Generally, a visual scene can be defined as a view that objects and other semantic surfaces are arranged in a meaningful way [42]. Scenes contain semantic components arranged in a spatial layout which can be observed at a variety of spatial scales (e.g., the up-close view of an office desk or the view of the entire office). Therefore, when building computational models to perform scene recognition, we need to consider the multi-scale property of scene images. Specifically, in this section, we first describe the basic network structure used in our exploration and then present the framework of multi-resolution CNNs.

A. Basic network structures

Deep convolutional networks have witnessed great successes in image classification and many effective network architectures have been developed, such as AlexNet [28], GoogLeNet [32], VGGNet [31], and ResNet [35]. As the dataset size of Places2 is much larger than that of ImageNet, we need to keep a good balance between recognition performance and computational cost when choosing network structure. In our experiment, we choose the inception architecture with batch normalization [19] (bn-inception) as our network structure. In addition to its good balance between accuracy and efficiency, inception architecture also leverages the idea of multi-scale processing when constructing inception module. Therefore, the inception architecture is a reasonable choice for constructing scene recognition networks.

As shown in Figure 2, the original bn-inception architecture starts with two convolutional layers and max pooling layers to transform 224×224 input images into 28×28 feature maps, whose sizes are relative small for the fast processing in the subsequent layers. Then, it contains ten inception layers, where two of them have stride of 2 and the rest have stride of 1. The size of feature map after these inception layers is changed to 7×7, and a global average pooling is used to aggregate these activations across spatial dimensions. Batch Normalization (BN) is applied to the activations of convolutional layers before they are fed into Rectified Linear Unit (ReLU) for non-linearity.

B. Two-resolution architectures

The proposed Multi-Resolution CNNs are decomposed into fine resolution and coarse resolution components in the current implementation. The coarse resolution CNN is the same with the normal bn-inception as specified in previous subsection, while fine resolution CNN shares a similar but deeper architecture.

Coarse resolution CNNs operate on image regions of size 224×224 and contain totally 13 layers with weights. The network structure of coarse resolution CNN is called as *normal bn-inception* since has the same structure as the original one in [19]. It captures visual appearance and structure at a relatively coarse resolution, focusing on describing objects at large scale. Therefore, some fine details may not be described well in such a coarse resolution. However, in natural images, there are many local objects, which play important roles for scene understanding. Hence, it requires to capture visual content in a finer resolution with focus on more details.

Fine resolution CNNs are developed for images of resolution 384×384 and operate on image regions of 336×336. As fine resolution CNN takes larger images as input, its depth can be increased. In the current implementation, to keep balance between speed and network capacity, we add three extra convolutional layers on top of inception layers, as illustrated in Figure 2. For these newly-added convolutional layers, the pad sizes are set as zeros and so the feature map size also becomes 7×7 before global average pooling. We call this network structure of fine resolution CNN as *deeper bn-inception*. Fine resolution CNNs aim to describe the image information and structure at finer scale, which allows to capture details.

These two-resolution CNNs take different resolution images as input and their receptive fields of the corresponding layers describe different regions of original images. They are designed to describe objects at different scales for scene understanding. Therefore, the prediction scores of CNNs from different resolutions are complementary to each other and we combine them by taking an arithmetic average.

Discussion. Although sharing similar ideas with common multi-scale training strategy [31], the proposed multi-resolution CNNs differ from it mainly on two aspects: (1) the input image sizes are different in our two-resolution architectures (224×224 and 336×336), but the input size is all the same in multi-scale training (only 224×224). (2) we design two distinct network structures in our MR architecture (normal bn-inception and deeper bn-inception) to handle different input sizes, while conventional multi scale training simply applies to a single network structure. Thanks to these differences, the proposed multi-resolution architecture is more suitable to capturing different level visual information for scene understanding. Moreover, the multi-resolution architecture is complementary to multi-scale training and can be easily combined with it as stated in next paragraph.

Training of multi-resolution CNNs. The training of multi-resolution CNNs are performed for each resolution independently. We train each CNN according to the common setup of [28], [31]. We use the mini-batch stochastic gradient descent algorithm to learn the network weights, where the batch size is set as 256 and momentum set to 0.9. The learning rate is initialized as 0.1 and decreases according to a fixed schedule determined by the dataset size and specified in Section V. Concerning data augmentation, the training images are resized as $N \times N$, where N is set as 256 for normal bn-inception and 384 for deeper bn-inception. Then, we randomly crop a $w \times h$ region at a set of fixed positions, where cropped width w and height h are picked from $\{N, 0.825N, 0.75N, 0.625N, 0.5N\}$. Then these cropped regions are resized as $M \times M$ for network training, where M is set as 224 for normal bn-inception and 336 for deeper bn-inception. Meanwhile, these crops undergo a horizontal flipping randomly. Our proposed cropping strategy is an efficient way to implement the scale jittering [31].

IV. Knowledge Guided Disambiguation

As analyzed above, several scene categories start to overlap with others in the large-scale datasets, such as Places2 [17]. The increasing number of scene categories causes the problem of label ambiguity, which makes the training of multi-resolution CNNs more challenging. In this section we propose two simple yet effective methods to handle the issue of label ambiguity by exploiting extra knowledge. Specifically, we first introduce the method of utilizing knowledge from confusion matrix and we then propose the second method which resorts to knowledge from extra networks.

A. Knowledge from confusion matrix

As the number of scene classes increases, the difference between scene categories becomes smaller and some scene classes are easily confused with others from visual appearance. A natural way to relieve this problem is to re-organize the scene class hierarchy and merge very similar classes into a super category. In order to merge similar classes, we need to come up with a solution to define the similarity between scene categories. Although it is possible to ask human annotator to determine which classes can be merged, it is a time-consuming work. Here we propose a simple yet effective way to automatically merge visually ambiguous scene categories.

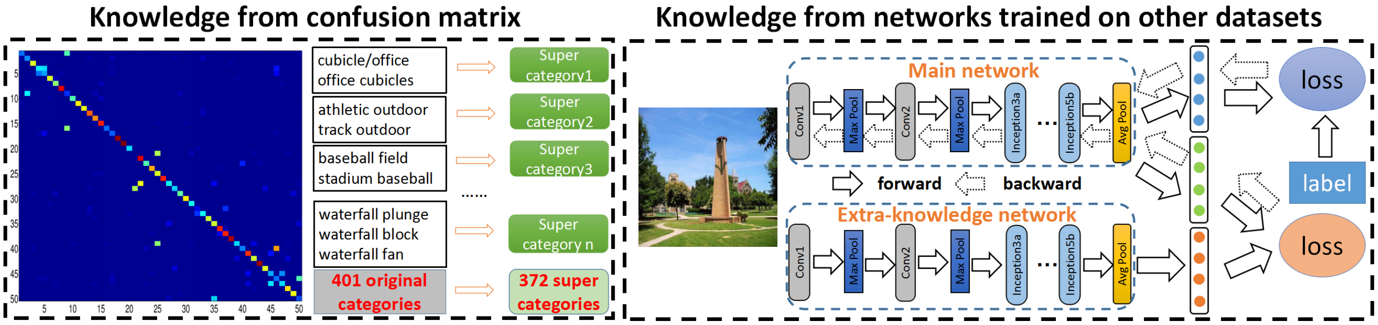

Fig. 3. **Knowledge guided disambiguation**: we propose two knowledge guided disambiguation methods to deal with the problem of overlapping labels. In the left, we utilize the knowledge of confusion matrix to merge similar scene classes into a super category and re-train CNNs on these relabeled datasets. In the right, we exploit the knowledge of extra networks trained on the other datasets to transform each image to a soft code, which can be used to guide the CNN training.

```
Algorithm 1: Merge similar classes into super category
  Data: Similarity matrix $\mathbf{S}$, threshold: $\tau$.
  Result: Merged classes: $\mathcal{S}$.
  - Initialization: $\mathcal{S} = \leftarrow \{S_1, S_2, \cdots, S_N\}$.
  while $max(\mathbf{S}) < \tau$ do
      1. Pick the maximum of similarity:
      $(i,j)^* \leftarrow \arg\max_{i,j} \mathbf{S}_{ij}$
      2. Merge the $i^{*th}$ and $j^{*th}$ classes into a single class
      : $\mathcal{S} = \mathcal{S} - \{S_{i^*}\} - \{S_{j^*}\} + \{(S_{i^*}, S_{j^*})\}$
      3. Update the similarity matrix by deleting $i^*$ and $j^*$
      rows and columns and adding a new row and column
      defined as $\frac{1}{2}(\mathbf{S}_i + \mathbf{S}_j)$
  end
  - Return merged classes: $\mathcal{S}$.
```

Specifically, we first train a deep model on the training dataset of Places2 with 401 classes. Then, we test the learned model on the validation dataset of Places2. The confusion matrix on the validation dataset reveals the fact that which scene pairs are easily confused with each other. Meanwhile, this confusion matrix also contains information on the similarity between each pair of scene category. Hence, we choose the confusion matrix to calculate the pairwise similarity of scene classes as follows:

$$\mathbf{S} = \frac{1}{2}(\mathbf{C} + \mathbf{C}^\top), \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{N\times N}$ is the confusion matrix, $\mathbf{C}_{ij}$ represents the probability of classifying i^{th} class as j^{th} class, N is the number of scene classes. This equation ensures the similarity measure is a symmetric metric.

After having similarity measure, we propose a bottom-up clustering algorithm to merge similar categories iteratively, as shown in Algorithm 1. At each iteration, we pick a pair of categories with the largest similarity and merge them into a super category. Then we update the similarity matrix $\mathbf{S}$ accordingly, by deleting i^{*th} and j^{*th} rows and columns and adding a new row and column defined as $\frac{1}{2}(\mathbf{S}_{i^*} + \mathbf{S}_{j^*})$, where $\mathbf{S}_{i^*}$ denotes the i^{*th} row vector of $\mathbf{S}$. This iteration repeats until there is no similarity value larger than τ. After this merging process, very similar scene categories are merged into a similar super category and all these images from these categories are supposed to have the same label, from which we will re-train a CNN for with smaller scene classes. In current implementation, the 401 scene classes from the Places2 dataset are re-organized as 372 super-categories. For testing these re-trained CNNs, in current implementation, we equally divide the probability of each super category into its sub categories. This simple strategy turns out to be effective in practice.

B. Knowledge from extra networks

In previous knowledge disambiguation method, we simply consider the similarity between scene classes and merge similar categories into a super category. However, this relabeling (merging) strategy treats all the images from the same class equally and ignores the difference contained in each single image. Intuitively, only part of images from these visually ambiguous classes are easily confused with each other and the other part may not. Hence, in this subsection, we propose to exploit knowledge from extra networks to incorporate the visual information of each single image into the relabeling procedure.

In order to consider the visual information of each single image in the relabeling procedure, a natural solution is to directly ask human with experience to relabel each image again. However, this solution is faced with two difficulties: (1) It will be time costly and require huge labor force, (2) It is hard to define the relabeling criteria to guide the human annotation. At the same time, publicly available CNNs trained on a relatively smaller and well-labeled dataset (e.g. ImageNet [20] or Places [17]) encode rich knowledge and can extract high-level semantics from raw images. Therefore, we may utilize these public models as a knowledge network to automatically relabel each image and treat their outputs as the soft labels of images.

Essentially, this soft label is a kind of distributed representation, which describes the scene content of each image with a distribution over common object classes or smaller subsets of scene categories. As shown in Figure 4, for instance, the content of *dinning room* could be described by distribution of common objects, where objects such as *dinning table* and *door* may dominate this distribution. For other scene category such as *office*, the objects of *screen* and *desktop computer*

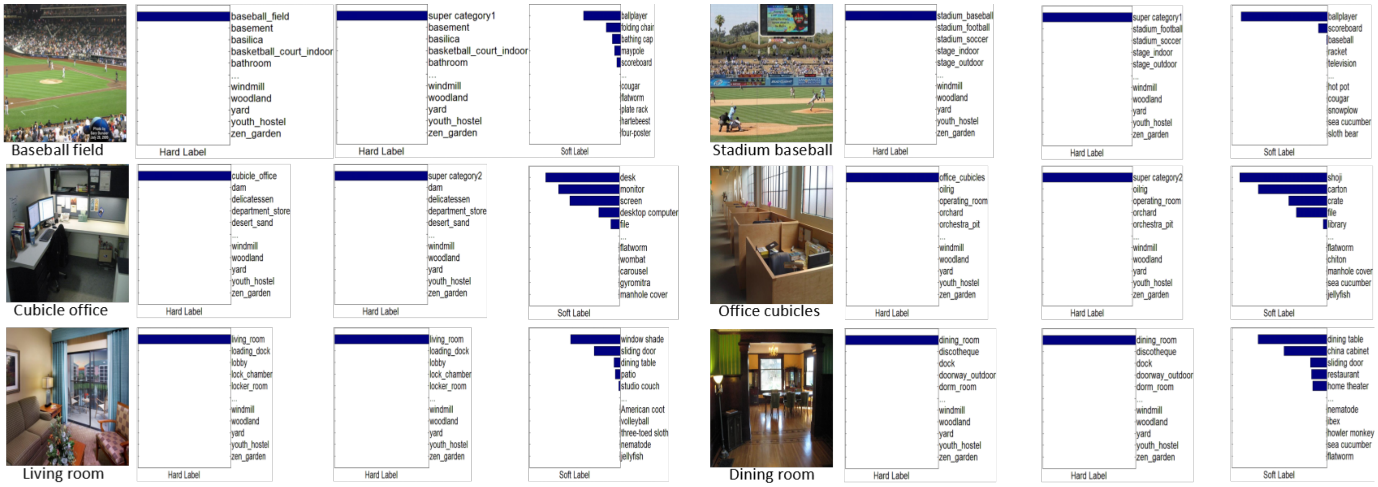

Fig. 4. **Hard and soft labels**: Several image examples with ground truth from the Places2 dataset. First, in the left histogram, we show the original hard labels provided by the dataset. Second, in the middle histogram, the hard labels are shown after merging visually ambiguous classes (our first disambiguation approach). In these examples, we see that classes of baseball field and stadium baseball, cubicle office and office cubicles, are merged into super category 1 and super category2. Finally, in the right histogram, we provide the soft labels produced by extra networks (our second disambiguation approach), where scene content is described by the distribution of common objects.

may have high probability mass. Utilizing this soft label to represent image content exhibit two main advantages: (1) For visually ambiguous classes, they typically share similar visual elements such as objects and background. So the soft labels of these classes will look similar and can encode the correlation of scene categories implicitly. (2) Compared with above label merging method, this soft label depends on the image content and may vary for different images. Normally, images from easily ambiguous classes may share similar but not identical soft labels. Hence, soft labels can still capture the subtle difference contained in each single image and is more informative than hard labels.

In current implementation, we consider the complementarity between groundtruth hard labels and soft labels from knowledge networks, and design a multi-task learning framework to utilize both labels to guide CNN training as shown in Figure 3. Specifically, during the training procedure, our CNNs predict both the original hard labels and the soft labels simultaneously, by minimizing the following objective function:

$$\ell(D) = -(\sum_{\mathbf{I}_i \in D} \sum_{k=1}^{K_1} \mathbb{I}(y_i = k) \log p_{i,k} + \lambda \sum_{\mathbf{I}_i \in D} \sum_{k=1}^{K_2} q_{i,k} \log f_{i,k}), \tag{2}$$

where D denotes the training dataset, $\mathbf{I}_i$ is the i^{th} image, y_i is its scene label (hard label), f_i is its soft code (soft label) produced by extra knowledge network, p_i is the output for hard code of image $\mathbf{I}_i$, q_i is the output for soft code of image $\mathbf{I}_i$, λ is a parameter balancing these two terms (set as 0.5 in experiment), and K_1 and K_2 are the dimension of hard label and soft label, respectively.

This multi-task objective function forces the training procedure to optimize the classification performance of original scene classification and imitate the knowledge network at the same time. This multi-task learning framework is able to improve generalization ability by exploiting the knowledge contained in extra networks as an inductive bias, and reduce the effect of over-fitting on the training dataset of Places2. As we shall see in Section V, this framework is able to improve the recognition performance of original multi-resolution CNNs.

V. Experiments

In this section, we describe the experimental settings and report the performance of our proposed method on the datasets of ImageNet [20], Places [1], Places2 [17], LSUN [43], MIT Indoor67 [22], and SUN397 [2]. We first describe our evaluation datasets and the implementation details. Then, we perform experiments to verify the effectiveness of multi-resolution CNNs on three datasets. After this, we conduct experiments to explore the effect of knowledge guided disambiguation on the dataset of Places2. We also report the performance of our method on two large-scale scene recognition challenges, namely Places2 challenge 2015 and LSUN challenge 2016. Meanwhile, we test the generalization ability of our learned models on the datasets of MIT Indoor67 [22] and SUN397 [2]. Finally, we give several examples that our methods fail to predict the correct label.

A. Large-scale datasets and implementation details

We first perform experiments on three large-scale image classification datasets to evaluate our proposed method, namely ImageNet [20], Places [1], and Places2 [17]. Due to the fact that the labels of testing data of these datasets are not available, we mainly evaluate our methods on their validation data.

ImageNet [20] is an object-centric dataset and the largest benchmark for object recognition and classification [1]. The dataset for ILSVRC 2012 contains 1,000 object categories. The training data contains around 1,300,000 images from these object categories. There are 50,000 images for validation

[1] http://image-net.org/

TABLE I
PERFORMANCE OF NORMAL BN-INCEPTION, DEEPER BN-INCEPTION, AND MULTI-RESOLUTION CNNS ON THE VALIDATION DATA FROM THE DATASETS OF IMAGENET, PLACES, AND PLACES2.

Method	ImageNet (top1/top5)	Places (top1/top5)	Places2 (top1/top5)
AlexNet [28]	40.7%/18.2%	50.0%/-	57.0%/-
VGGNet-16 [31]	27.0%/8.8%	39.4%/11.5%	52.4%/-
Normal BN-Inception	24.7%/7.2%	38.1%/11.3%	48.8%/17.4%
Deeper BN-Inception	23.7%/6.6%	37.8%/10.7%	48.0%/16.7%
Multi-resolution CNN	21.8%/6.0%	36.4%/10.4%	47.4%/16.3%

dataset and 100,000 images for testing. The evaluation measure is based on top-5 error, where algorithms will produce a list of at most 5 object categories to match the ground truth.

Places [1] is a large-scale scene-centric dataset [2]. Places dataset selects 205 common scene categories (referred to as Places205). The training dataset contains around 2,500,000 images from these categories. For the training set, each scene category has the minimum 5,000 and maximum 15,000 images. The validation set contains 100 images per category (a total of 20,500 images) and the testing set contains 200 images per category (a total of 41,000 images). The evaluation criteria of Places is also based on top-5 error.

Places2 [17] is extended from the Places dataset and the largest scene recognition dataset currently [3]. In total, Places2 contains more than 10 million images comprising more than 400 unique scene categories. The dataset includes 5000 to 30,000 training images per class, consistent with real-world frequencies of occurrence. In the Places2 challenge 2015 (held in conjunction with ImageNet large-scale visual recognition challenge), it contains 401 scene categories. The training dataset of Places2 has around 8,100,000 images. The validation set contains 50 images per category and the testing set contains 950 images per category. Due to the much larger size, scene recognition on Places2 is more challenging than other datasets.

The training details of our proposed method on these three datasets are similar, as specified in Section III. The only difference is the iteration number due to the different sizes of training data for these datasets. Specifically, on the ImageNet and Places datasets, we decrease learning rate every 200,000 iterations and the whole training procedure stops at 750,000 iterations, while on the Places2 dataset, learning rate is decreased every 350,000 iterations and the whole training process ends at 1,300,000 iterations. We use the multi-GPU extension [44] of Caffe [45] toolbox for our CNN training [4]. For testing our learned models, we use the common 5 crops (4 corners and 1 center) and their horizontal flipping for each image at a single scale. Totally, there are 10 crops for each image.

B. Evaluation on multi-resolution CNNs

We begin our experiment study with exploring the effectiveness of multi-resolution CNNs on the validation set of ImageNet, Places, and Places2. Specifically, we study three architectures: (1) normal BN-Inception, which is trained from 256 × 256 images, (2) deeper BN-Inception, which has a deeper structure and is trained from 384 × 384 images, (3) multi-resolution CNN, which is combination of normal BN-Inception and deeper BN-Inception and the fusion weights are set to be equal to each other.

TABLE II
PERFORMANCE OF DIFFERENT KNOWLEDGE GUIDED DISAMBIGUATION TECHNIQUES ON THE DATASET OF PLACES2.

Method	Places2 Val
(A0) Normal BN-Inception (256 × 256)	17.4%
(A1) Normal BN-Inception + object networks	17.4%
(A2) Normal BN-Inception + scene networks	16.7%
(A3) Normal BN-Inception + confusion matrix	17.3%
Fusion of (A0) and (A1)	16.7%
Fusion of (A0) and (A2)	16.3%
Fusion of (A0) and (A3)	16.6%
(B0) Deeper BN-Inception (384 × 384)	16.7%
(B1) Deeper BN-Inception + object networks	16.3%
(B2) Deeper BN-Inception + scene networks	16.1%
Fusion of (B0) and (B1)	15.9%
Fusion of (B0) and (B2)	15.8%

The results are summarized in Table I. First, from comparison of normal BN-Inception and deeper BN-Inception, we conclude that CNNs trained from fine resolution images (384 × 384) are able to yield better performance than those trained from coarse resolution images (256 × 256) on all these datasets. This superior performance may be ascribed to the fact that fine resolution images contain more rich information of visual content and local details. In addition, the deeper BN-Inception is able to exhibit higher modeling capacity and capture scene content more effectively. Second, we take an arithmetic average over the scores of normal BN-Inception and deeper BN-Inception as the results of multi-resolution CNNs. This simple fusion can further boost the recognition performance on three datasets. This improvement indicates that the information captured by CNNs from different resolution images are complementary to each other. Finally, we compare our mulit-resolution CNNs with other baselines (AlexNet and VGGNet-16) on three datasets and our approach outperforms these baselines by a large margin. It is worth noting that our multi-resolution CNN is a general learning framework that can be applied to existing network structures to enhance their modeling capacity.

[2] http://places.csail.mit.edu/

[3] http://places2.csail.mit.edu/

[4] https://github.com/yjxiong/caffe

TABLE III
PERFORMANCE OF DIFFERENT TEAMS AT PLACES2 CHALLENGE 2015.

Rank	Team	Places2 Test	Places2 Val
1	WM [9]	**16.9%**	15.7%
2	SIAT_MMLAB (A2+B0)	17.6%	16.2%
2	SIAT_MMLAB (A0+A1+A2+A3+B0)	17.4%	15.8%
-	Post submission (B0+B1+B2)	-	**15.5%**
3	Qualcomm	17.6%	-
4	Trimps-Soushen	18.0%	-
5	NTU-Rose	19.3%	-

TABLE IV
PERFORMANCE OF DIFFERENT PRE-TRAINED MODELS ON THE VALIDATION SET OF LSUN CLASSIFICATION DATASET.

Pre-trained Model	Top1 Accuracy
(A0) Normal BN-Inception (256×256)	89.9%
(A1) Normal BN-Inception + object networks	90.1%
(A2) Normal BN-Inception + scene networks	90.4%
(B0) Deeper BN-Inception (384×384)	90.5%
(B1) Deeper BN-Inception + object networks	90.7%
(B2) Deeper BN-Inception + scene networks	90.9%
(A0+B0)	91.0%
Fusion all	91.8%

C. Evaluation on knowledge guided disambiguation

After the investigation of the effectiveness of multi-resolution CNNs, we now turn to study the effect of our proposed knowledge guided disambiguation techniques in Section IV. To handle the issue of label ambiguity existed scene recognition, we proposed two disambiguation techniques, one based on the knowledge of confusion matrix on the validation dataset, and the other based on the knowledge from extra networks. As the label ambiguity is an important issue for large-scale scene recognition, we perform experimental exploration on the Places2 dataset in this subsection.

In the first knowledge guided disambiguation technique, according to the confusion matrix, we merge 401 scene categories into 372 super categories. The results are shown in Table II. We see that for normal BN-Inception network, the performance of utilizing knowledge from confusion matrix is slightly better than the normal BN-Inception. This result is a little bit surprising, as we use less category information but obtain better performance. This result indicates that label ambiguity may leads to the problem of over-fitting with more subtle information to distinguish easily confused categories (e.g. baseball field vs. stadium baseball). But these subtle difference may not generalize well on testing data and so decrease the recognition performance.

In the second knowledge guided disambiguation technique, we utilize two extra networks: one trained on the ImageNet dataset (object network) and one trained on the Places (scene network). We use the outputs of these networks as soft labels to guide the training of CNNs. The results are reported in Table II. For normal BN-Inception architecture, the object network guided CNN achieves the same performance with the original one. The scene network guided CNN obtains much better performance (16.7% vs. 17.4%). For deeper BN-Inception architecture, the performance can be improved for both object and scene network guided CNNs. These results imply that exploitation of knowledge from extra networks is an effective way to regularize the training of the original networks and improve the generalization ability. Meanwhile, we notice that the soft labels from scene networks outperform those from object networks, which may be ascribed to the fact that the scene classes from Places are more correlated with Places2 classes than those object classes from ImageNet.

Finally, we perform model fusion with normally trained CNNS and knowledge guided CNNs. From these results, we see that those knowledge guided CNNs are complementary to those normally trained CNNs. For normal BN-Inception architecture, the best combination of (A0) and (A2) is able to reduce the top-5 error to 16.3% from 17.4%. With deeper BN-Inception network, the best combination of (B0) and (B2) achieves the top-5 error of 15.8% compared with original top-5 error of 16.7%. These better fusion results indicate that our proposed knowledge guided disambiguation techniques can not only improve the performance the original models, but also provide complementary models to build a strong model ensemble.

D. Results at Places2 challenge 2015

After the separate study of multi-resolution CNNs and knowledge guided disambiguation, we are ready verify its effectiveness on large-scale scene recognition challenge. In this subsection we present the results of our method on the Places2 challenge 2015. Places2 challenge is the largest scene recognition challenge and held in conjuction with the ImageNet large-scale visual recognition challenge (ILSVRC) [21].

The challenge results are summarized in Table III and our team secures the 2^{nd} place. Compared the winner method [9], our performance is lower by 0.5% in top-5 error. During test phase, there is a big difference between our approach and winner method, where they exploited the multi-scale cropping strategy while we simply choose the single-scale cropping method. In addition, it is worth noting that our submission did not contain the best model architecture B2 due to the challenge deadline. After the challenge, we finish the training of B2 network and it achieves better performance on the validation dataset. Finally, we achieve the performance of 15.5% top-5 error on the validation set, which is a little bit better than that of the winner method (15.7%).

E. Results at LSUN challenge 2016

In this subsection, we report the performance of our method on another important scene recognition challenge, namely LSUN. Large-Scale Scene Understanding (LSUN) challenge aims to provide another benchmark for scene classification and understanding [5]. The LSUN classification dataset [43] contains 10 scene categories, such as dining room, bedroom, chicken, outdoor church, and so on. For training data, each category contains a huge number of images, ranging from

[5] http://lsun.cs.princeton.edu

TABLE V
PERFORMANCE OF DIFFERENT TEAMS AT LSUN CHALLENGE 2016.

Rank	Team	Year	Top1 Accuracy
1	SIAT_MMLAB	2016	**91.6%**
2	SJTU-ReadSense	2016	90.4%
3	TEG Rangers	2016	88.7%
4	ds-cube	2016	83.0%
1	Google	2015	91.2%

around 120,000 to 3,000,000. The validation data includes 300 images and the test data has 1000 images for each category. The evaluation of LSUN classification challenge is based on the top-1 accuracy.

In order to verify the effectiveness of our proposed multi-resolution CNN and knowledge guided disambiguation strategy, we choose to transfer these learned representations on Places2 dataset to the classification task of LSUN challenge. Specifically, to reduce the computational cost and keep a balance between different classes, we randomly sample 100,000 images for each scene category as our training data. Then, we use these learned CNNs on the Places2 dataset as pre-training models and fine tune network parameters on the LSUN dataset. The learning rate is initialized as 0.1 and it is decreased to its $\frac{1}{10}$ every 60,000 iterations, where batch size is set as 256. The whole training process stops at 180,000 iterations. During the test phase, following the common techniques, we crop 5 regions and their horizontal flipping, and use 3 different scales for each image. We take an average over the prediction scores of these different crops as the final recognition result of this image.

We first report the performance of different pre-trained models on the validation set of LSUN dataset and the results are reported in Table IV. First, comparing the performance of CNNs at different resolutions, we find that the deeper BN-Inception networks learned on finer resolution images can yield better performance than the normal BN-Inception networks (89.9% vs. 90.5%). Second, considering the strategy of knowledge guided disambiguation, both object and scene guided CNNs are capable of bringing improvement (around 0.5%) over non-guided CNNs. Finally, we fuse the predictions of multiple networks and obtain the final performance of 91.8% on the validation set of LSUN dataset.

We also provide the results of our method (fusion all) on the test set of LSUN dataset in Table V and compare with other teams at this challenge. Our SIAT_MMLAB team obtains the performance of 91.6% and secures the 1^{st} place at this challenge, which demonstrates the effectiveness of our proposed solution for scene recognition. Importantly, our performance is better than the winner of LSUN 20015 (Google) by 0.4%, which also used a similar Inception architecture, but lacked considering the multi-resolution structure and knowledge guided disambiguation strategy.

F. Generalization analysis

The previous experiments have demonstrated the effectiveness of our proposed method on the large-scale datasets in both settings of training from scratch (Places2) and adaption with fine tuning (LSUN). In this subsection, we aim to test the generalization ability of our learned models on other relatively small scene recognition datasets. It should be noted that although the sizes of these datasets are relatively small, they have been uesed as standard scene recognition benchmarks for a few years and many competitive methods have reported performance on these datasets. Specifically, we choose two scene recognition datasets: MIT Indoor67 [22] and SUN397 [2].

TABLE VI
PERFORMANCE COMPARISON OF TRANSFERRED REPRESENTATIONS OF OUR MODEL WITH OTHER METHODS ON THE MIT67 AND SUN397 DATASETS.

Model	MIT Indoor67	SUN397
ImageNet-VGGNet-16 [31]	67.7%	51.7%
Places205-AlexNet [1]	68.2%	54.3%
Places205-GoogLeNet [46]	74.0%	58.8%
DAG-VggNet19 [8]	77.5%	56.2%
Places205-CNDS-8 [47]	76.1%	60.7%
Ms-DSP [48]	78.3%	59.8%
Places205-VGGNet-16 [49]	81.2%	66.9%
LS-DHM [46]	83.8%	67.6%
Multiple Models [50]	86.0%	70.7%
Three [51]	86.0%	70.2%
Places2-Deeper-BN-Inception	**86.7%**	**72.0%**

The MIT Indoor67 [22] contains 67 indoor-scene categories and has a total of 15,620 images, with at least 100 images per category. Following the original evaluation protocol, we use 80 images from each category for training, and another 20 images for testing. The SUN397 [2] has a large number of scene categories by including 397 categories and totally 108,754 images. Each category has at least 100 images. We follow the standard evaluation protocol provided in the original paper. We test the our method with each category having 50 training and 50 test images. The partitions are fixed and publicly available from [2]. Finally the average classification accuracy of ten different tests is reported.

In this experiment, we treat the learned models (B2) as generic feature extractors without fine tuning on the target dataset. Specifically, the test images are first resized as 384×384. We then crop image regions of different scales (384×384, 346×346, and 336×336) from the input images. After this, these image regions are resized as 336×336 and fed into CNNs for feature extraction. We utilize the activation of global pooling as the global representation. These global representations of different regions are averaged and normalized with ℓ_2-norm. For classifier, we use the linear SVM with LIBSVM implementation [52].

The experimental results are summarized in Table VI. We compare the transfered representations of our model trained on the Places2 dataset with other deep models (e.g. VGGNet [31] and GoogLeNet [32]) trained on different datasets (e.g. Places and ImageNet). From these results, we see that our learned representations are more generic and achieve better performance. To the best of our knowledge, the performance of 86.7% on the MIT Indoor67 and 72.0% on the SUN397 are the best ones for both datasets, which advance the state of the art substantially. We believe such good performance is valuable to the scene recognition community and the future recognition algorithm can be built on our pre-trained models.

G. Failure case analysis

Finally, we visualize some examples that our method fails to predict the correct labels on the datasets of Places2 and LSUN. These examples are illustrated in Figure 5. From these examples, we notice that some scene classes are easily confused with others. For the Places2 dataset, the categories of supermarket, pet-store, toyshop look very similar from the outdoor appearance. The classes of *downtown*, *building*, and *skyscraper* may co-occur in many images. Thus, sometimes scene recognition is a kind of multi-label classification problem and single label is not enough to describe the scene content. For the dataset of LSUN, the classes of *bridge* and *tower* are easily confused with each other, as they look quite similar in some cases. Also, the category of *conference room* is sometimes confused with the *classroom* category due to similar spatial layout and common objects. Overall, from these failure cases, we can see that scene recognition is still an challenging problem and label ambiguity is a important issue in the large-scale scene recognition, which still needs to to be further explored in the future.

VI. Conclusions

In this paper, we have studied the problem of scene recognition on the large-scale datasets such as Places, Places2, and LSUN. Large-scale scene recognition is faced with two major issues: visual inconsistence (large intra-class variation) and label ambiguity (small inter-class variation). We designed two techniques to address these problems accordingly: multi-resolution CNNs are able to capture visual information from different scales and knowledge guided disambiguation techniques exploit extra knowledge to relabel images and improve the generalization ability of learned models.

We conducted experiments on three large-scale image classification datasets to demonstrate the effectiveness of our proposed approach. In addition, our method took part in two major scene recognition challenges, and we achieved the 2^{nd} place at the Places2 challenge 2015 and 1^{st} place at the LSUN challenge 2016. The top performance further verify the superior performance of our method over previous works. Finally, we also tested the generalization ability our learned models on other relatively small but competitive datasets, where our learned representations obtained the current state-of-the-art performance on the datasets of MIT Indoor67 and SUN397.

Scene recognition is essentially a multi-label classification problem. We will consider annotation with multi-labels in the future. We need to better take into account of the label correlations and may exploit other semantic concepts like objects for scene understanding in still images. Meanwhile, we could also come up with the scene-centric CNN architectures to capture both the global layout and local details for scene recognition.

References

[1] B. Zhou, À. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014, pp. 487–495.

[2] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010, pp. 3485–3492.

[3] F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005, pp. 524–531.

[4] L. Zhang, X. Zhen, and L. Shao, "Learning object-to-class kernels for scene classification," *IEEE Trans. Image Processing*, vol. 23, no. 8, pp. 3241–3253, 2014.

[5] Z. Zuo, G. Wang, B. Shuai, L. Zhao, Q. Yang, and X. Jiang, "Learning discriminative and shareable features for scene classification," in *ECCV*, 2014, pp. 552–568.

[6] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *ECCV*, 2014, pp. 392–407.

[7] L. Xie, J. Wang, B. Guo, B. Zhang, and Q. Tian, "Orientational pyramid matching for recognizing indoor scenes," in *CVPR*, 2014, pp. 3734–3741.

[8] S. Yang and D. Ramanan, "Multi-scale recognition with DAG-CNNs," in *ICCV*, 2015, pp. 1215–1223.

[9] L. Shen, Z. Lin, and Q. Huang, "Learning deep convolutional neural networks for places2 scene recognition," *CoRR*, vol. abs/1512.05830, 2015.

[10] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[11] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, no. 2, pp. 169–191, 2003.

[12] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[13] L. Wang, Z. Wang, W. Du, and Y. Qiao, "Object-scene convolutional neural networks for event recognition in images," in *CVPR Workshops*, 2015, pp. 30–35.

[14] Y. Xiong, K. Zhu, D. Lin, and X. Tang, "Recognize complex events from static images by fusing deep channels," in *CVPR*, 2015, pp. 1600–1609.

[15] Y. Wang, J. Song, L. Wang, L. V. Gool, and O. Hilliges, "Two-stream SR-CNNs for action recognition in videos," in *BMVC*, 2016.

[16] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *IEEE Trans. Image Processing*, vol. 23, no. 2, pp. 810–822, 2014.

[17] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva, "Places2: A large-scale database for scene understanding," *Arxiv*, 2015.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[20] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[22] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *CVPR*, 2009, pp. 413–420.

[23] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006, pp. 2169–2178.

[24] S. N. Parizi, J. G. Oberlin, and P. F. Felzenszwalb, "Reconfigurable models for scene recognition," in *CVPR*, 2012, pp. 2775–2782.

[25] M. Pandey and S. Lazebnik, "Scene recognition and weakly supervised object localization with deformable part-based models," in *ICCV*, 2011, pp. 1307–1314.

[26] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *ECCV*, 2012, pp. 73–86.

[27] M. Juneja, A. Vedaldi, C. V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," in *CVPR*, 2013, pp. 923–930.

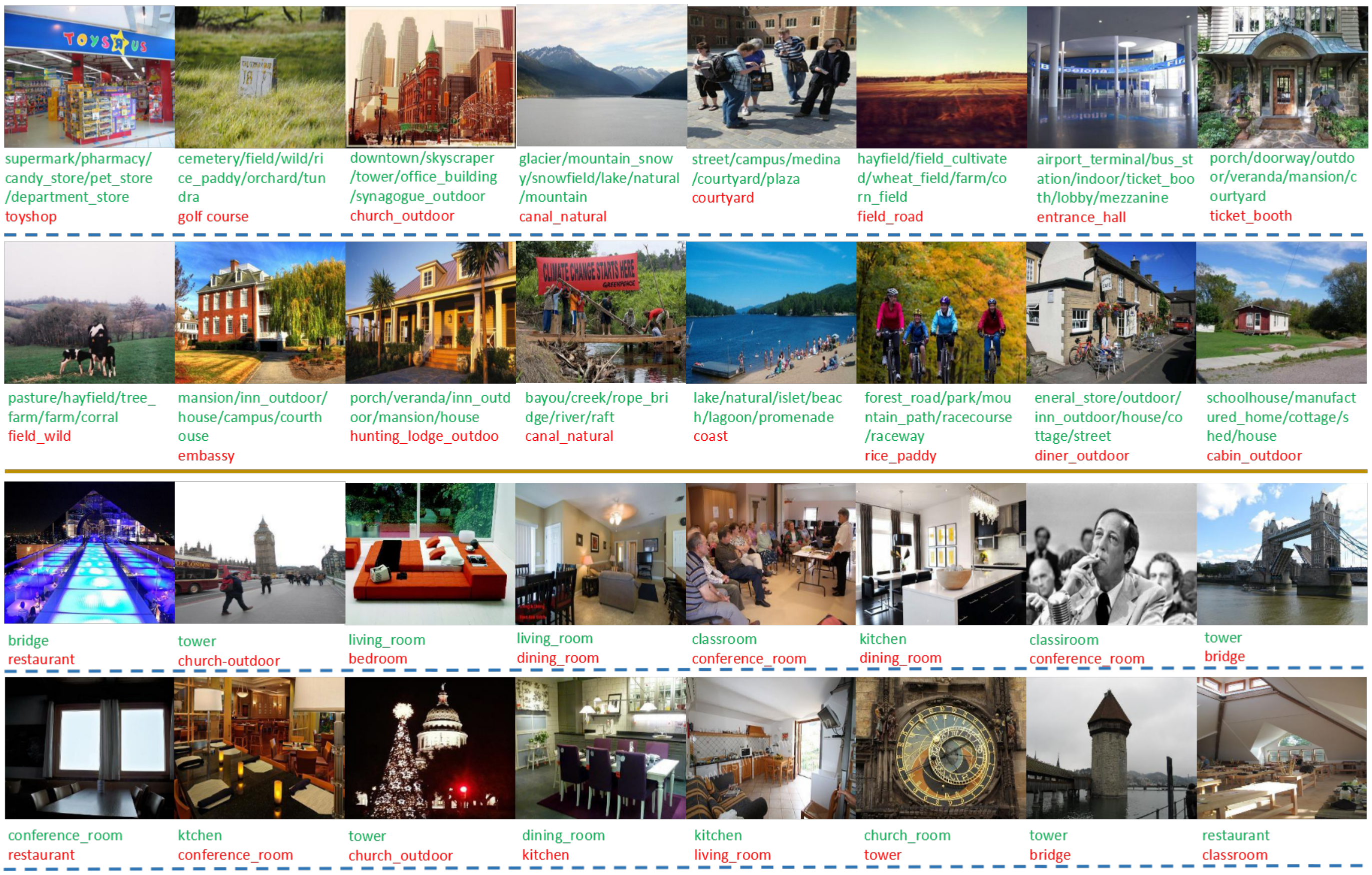
supermark/pharmacy/candy_store/pet_store/department_store
toyshop
cemetery/field/wild/rice_paddy/orchard/tundra
golf course
downtown/skyscraper/tower/office_building/synagogue_outdoor
church_outdoor
glacier/mountain_snowy/snowfield/lake/natural/mountain
canal_natural
street/campus/medina/courtyard/plaza
courtyard
hayfield/field_cultivated/wheat_field/farm/corn_field
field_road
airport_terminal/bus_station/indoor/ticket_booth/lobby/mezzanine
entrance_hall
porch/doorway/outdoor/veranda/mansion/courtyard
ticket_booth
pasture/hayfield/tree_farm/farm/corral
field_wild
mansion/inn_outdoor/house/campus/courthouse
embassy
porch/veranda/inn_outdoor/mansion/house
hunting_lodge_outdoo
bayou/creek/rope_bridge/river/raft
canal_natural
lake/natural/islet/beach/lagoon/promenade
coast
forest_road/park/mountain_path/racecourse/raceway
rice_paddy
eneral_store/outdoor/inn_outdoor/house/cottage/street
diner_outdoor
schoolhouse/manufactured_home/cottage/shed/house
cabin_outdoor
bridge
restaurant
tower
church-outdoor
living_room
bedroom
living_room
dining_room
classroom
conference_room
kitchen
dining_room
classiroom
conference_room
tower
bridge
conference_room
restaurant
ktchen
conference_room
tower
church_outdoor
dining_room
kitchen
kitchen
living_room
church_room
tower
tower
bridge
restaurant
classroom

Fig. 5. Examples of images that our method fail to predict the correct labels with 5 guesses. In the top rows, we show 16 failure cases (under **top-5** evaluation) on the validation set of the Places2 dataset. The predicted labels (in green) are sorted according to their confidence score and the correct label is labeled in red. In the bottom rows, we give 16 examples that our method fail to predict correct labels (under **top-1** evaluation) on the validation set of the LSUN dataset. Our predicted label is marked with green color and the ground truth with red color.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1106–1114.

[29] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014, pp. 818–833.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014, pp. 346–361.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *ICCV*, 2015, pp. 1026–1034.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[36] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[37] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *CoRR*, vol. abs/1412.6550, 2014.

[38] S. Gupta, J. Hoffman, and J. Malik, "Cross modal distillation for supervision transfer," in *CVPR*, 2016, pp. 2827–2836.

[39] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *ICCV*, 2015, pp. 4068–4076.

[40] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *CVPR*, 2016, pp. 2718–2726.

[41] C. Bucila, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *SIGKDD*, 2006, pp. 535–541.

[42] A. Oliva, "Scene perception," *Encyclopaedia of Perception*, 2009.

[43] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "LSUN: construction of a large-scale image dataset using deep learning with humans in the loop," *CoRR*, vol. abs/1506.03365, 2015.

[44] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *ECCV*, 2016.

[45] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *CoRR*, vol. abs/1408.5093.

[46] S. Guo, W. Huang, and Y. Qiao, "Locally-supervised deep hybrid model for scene recognition," *CoRR*, vol. abs/1601.07576, 2016.

[47] L. Wang, C. Lee, Z. Tu, and S. Lazebnik, "Training deeper convolutional networks with deep supervision," *CoRR*, vol. abs/1505.02496, 2015.

[48] B. Gao, X. Wei, J. Wu, and W. Lin, "Deep spatial pyramid: The devil is once again in the details," *CoRR*, vol. abs/1504.05277, 2015.

[49] L. Wang, S. Guo, W. Huang, and Y. Qiao, "Places205-vggnet models for scene recognition," *CoRR*, vol. abs/1508.01667, 2015.

[50] G. Xie, X. Zhang, S. Yan, and C. Liu, "Hybrid CNN and dictionary-based models for scene recognition and domain adaptation," *CoRR*, vol. abs/1601.07977, 2016.

[51] L. Herranz, S. Jiang, and X. Li, "Scene recognition with cnns: Objects, scales and dataset bias," in *CVPR*, 2016, pp. 571–579.

[52] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM TIST*, vol. 2, no. 3, p. 27, 2011.