Learning Structured Inference Neural Networks with Label Relations

Hexiang Hu[1,2], Guang-Tong Zhou[1], Zhiwei Deng[1], Zicheng Liao[2], and Greg Mori[1]

[1]School of Computing Science, Simon Fraser University , Burnaby, BC, Canada
[2]College of Computer Science and Technology, Zhejiang University , Hangzhou, Zhejiang, China
{hexiangh, gza11, zhiweid}@sfu.ca, zliao@zju.edu.cn, mori@cs.sfu.ca

Abstract

Images of scenes have various objects as well as abundant attributes, and diverse levels of visual categorization are possible. A natural image could be assigned with fine-grained labels that describe major components, coarse-grained labels that depict high level abstraction, or a set of labels that reveal attributes. Such categorization at different concept layers can be modeled with label graphs encoding label information. In this paper, we exploit this rich information with a state-of-art deep learning framework, and propose a generic structured model that leverages diverse label relations to improve image classification performance. Our approach employs a novel stacked label prediction neural network, capturing both inter-level and intra-level label semantics. We evaluate our method on benchmark image datasets, and empirical results illustrate the efficacy of our model.

1. Introduction

Standard image classification is a fundamental problem in computer vision – assigning category labels to images. It can serve as a building block for many different computer vision tasks including object detection, visual segmentation, and scene parsing. Recent progress in deep learning [17, 28, 29, 30] significantly improved classification performance on large scale image datasets [24, 36, 1, 19]. Approaches typically assume image labels to be semantically independent and adapt either a multi-class or binary classifier to label images. In recent work [2, 4], deep learning methods that take advantage of label relations have been proposed to improve image classification performance.

However, in realistic settings, these label relationships could form a complicated graph structure. Take Figure 1 as an example. Various levels of interpretation could be formed to represent such an image. This image of a *baseball* scene could be described as an *outdoor* image at coarse level, or with a more concrete concept such as *sports field*, or with even more fine-grained labels such as *batter's box* and objects such as *grass*, *bat*, *person*.

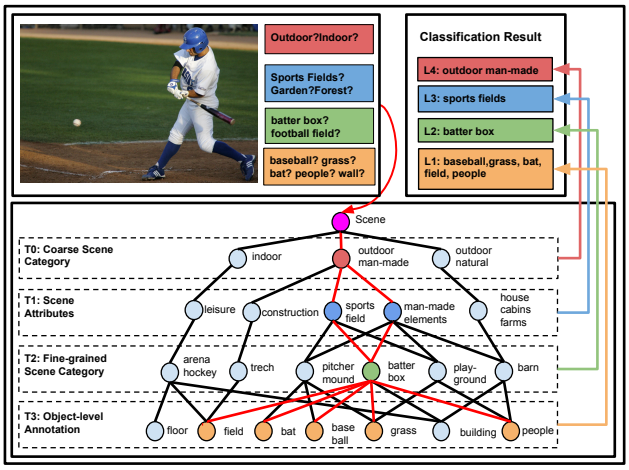

Figure 1. This image example has visual concepts at various levels, from sports field at high level to baseball and person at lower level. Our model leverages label relations and jointly predicts layered visual labels from an image using a structured inference neural network. In the graph, colored nodes correspond to the labels associated with the image, and red edges encode label relations.

Models that incorporate semantic label relationships could be utilized to generate better classification results. The desiderata for these models include the ability to model label-label relations such as positive or negative correlation, respect multiple concept layers obtainable from sources such as WordNet, and to handle partially observed label data – given a subset of accurate labels for this image, infer the remaining missing labels.

The contribution of this paper is in developing a structured inference neural network that permits modeling complex relations between labels, ranging from hierarchical to within-layer dependencies. We do this by defining a net-

work in which a node is activated if its corresponding label is present in an image. We introduce stacked layers among these label nodes. These encode layer-wise connectivity among label classification scores, representing dependency from top-level coarse labels to bottom-level fine-grained labels. Activations are propagated bidirectionally and asynchronously on the label relation graph, passing information about the labels within or across concept layers to refine the labeling for the entire image.

We have evaluated our method on three image classification datasets (AWA dataset [19], NUS-WIDE dataset [1] and SUN397 dataset [36]). Experimental results show a consistent and significant performance gain with our structured label relation model compared with baseline and related methods.

2. Related Work

Multi-level labeling of images has been addressed in a number of frameworks. In this section we review relevant work within probabilistic, max-margin, multi-task, and deep learning.

Structured label prediction with external knowledge: Structured prediction approaches exist [31, 33], in which a set of class labels are predicted jointly under a fixed loss function. Traditional approaches learn graph structure as well as associated weights that best explain the training data (*e.g.*, [3]). When external knowledge of label relations (*e.g.*, a taxonomy) is available, it is beneficial to integrate this knowledge to guide the traditional supervised learning systems. For example, Grauman *et al*. [7] and Hwang *et al*. [10] took the WordNet category taxonomy to improve visual recognition. Johnson *et al*. [13] and McAuley and Leskovec [21] used metadata from a social network to improve image classification. Ordonez *et al*. [23] leveraged associated image captions (words of "naturalness") to estimate entry-level labels of visual objects.

Multi-label classification with label relations: Traditional multi-label classification cannot avoid predicting an image as both *cat* and *dog*, or an image as *carnation* but not *flower*. Using external knowledge of label relations, Deng *et al*. [2] proposed a representation, the HEX graph, to express and enforce exclusion, inclusion and overlap relations between labels in multi-label classification. This model was further extended for "soft" label relations using the Ising model by Ding *et al*. [4].

Structured model with convolutional neural networks (CNNs): Structured deep models extend traditional CNNs to applications of structured label prediction, for which the CNN model is found insufficient to learn implicit constraints or structures between labels. Structured deep learning therefore jointly learns a structured model with the CNN framework. For example, for human pose estimation, Tompson *et al*. [32] take the CNN predictions as unary potentials for body parts and feed them to a MRF-like spatial model, which further learns pairwise potentials of part relations. Schwing and Urtasun [27] proposed a structured deep network by concatenating a densely connected MRF model to a CNN for semantic image segmentation, in which the CNN provides unary potentials as the MRF model imposes smoothness.

Our work combines these three lines of work. We take the WordNet taxonomy as our external knowledge, expressing it as a label relation graph, and learning the structured labels within a deep network framework. Our contribution is in proposing a learning and inference algorithm that facilitates knowledge passing in the deep network based on label relations.

Multi-task joint learning: Multi-task learning follows the same spirit of structured label prediction, with the distinction that the outputs of multiple (different but related) tasks are estimated. Common jointly modeled tasks include segmentation and detection [18, 34], segmentation and pose estimation [15], or segmentation and object classification [20]. An emerging topic of joint learning is in image understanding and text generation by leveraging intra-modal correspondences between visual and human language [16, 14].

Our work can be naturally extended to multi-task learning, for which each layer of our model represents one task and the labels do not necessarily form a layered structure. Notably, we can improve existing multi-task learning methods by importing knowledge of intra-task label relations.

3. Method

Our model jointly classifies image in a layered label space with external label relations. The goal is to leverage the label relations to improve inference over the layered visual concepts.

We build our model on top of state-of-the-art deep learning platform: given an image, we first extract CNN features from Krizhevsky *et al*. [17] as visual activation at each concept layer. Concept layers are stacked from fine-grained level to coarser levels. Label relations are defined between consecutive layers and form a layered graph. Inference over the label relation graph is inspired by the recent success of Recurrent Neural Network (RNN) [9, 26], where we treat each concept layer as a timestep of RNN. We connect neighboring timesteps to reflect the inter-layer label relations, while capturing intra-layer relations within each timestep. The label activations are propagated bidirectionally and asynchronously in the label relation graph to refine labeling for the given image. Figure 2 shows an overview of our classification pipeline.

We denote the collection of training images as $\{I^i\}_{i=1}^N$, each with ground-truth label in every concept layer. We denote the labels of image I^i as $\{y_t^i\}_{t=1}^T$, where T is the

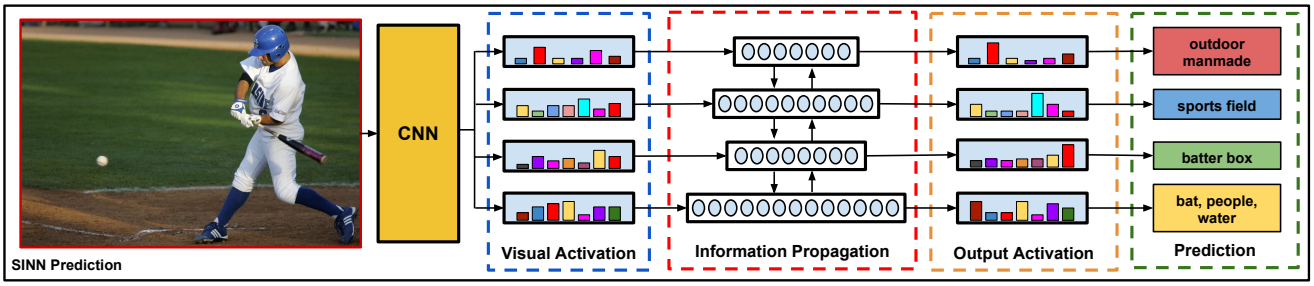

Figure 2. The label prediction pipeline. Given an input image, we extract CNN features at the last fully connected layer as activation (in blue box) at different visual concept layers. We then propagate the activation information in a label (concept) relation graph through our structured inference neural network (in red box). The final label prediction (in green box) is made from the output activations (in yellow box) obtained after the inference process.

total number of concept layers. And each concept layer t has n_t labels. The CNN framework of Krizhevsky *et al.* [17] transforms each image I^i into a 4096-dimensional feature vector, denoted as $CNN(I^i)$.

3.1. The Learning Framework

It is straightforward to build an image classification model, by adding a loss layer on top of the CNN features for each concept layer. Specifically, the activations on concept layer t are computed as

$$x_t^i = W_t \cdot CNN(I^i) + b_t, \tag{1}$$

where $W_t \in \mathbb{R}^{n_t \times 4096}$ and $b_t \in \mathbb{R}^{n_t \times 1}$ are linear transformation parameters and biases to classify the n_t labels at concept layer t. Note that $x_t^i \in \mathcal{R}^{n_t \times 1}$ provides visual activation depending purely on the visual image I^i. To generate label-specific probabilities, we can simply apply a sigmoid function (*i.e.*, $\sigma(z) = \frac{1}{1+e^{-z}}$) on the elements of x_t^i.

This classification model does not accommodate label relations within or across concept layers. To leverage the benefit of label relations, we adopt an RNN-like inference framework. In the following, we first describe a top-down inference model, then a bidirectional inference model, and lastly propose our Structured Inference Neural Network, the SINN.

3.1.1 Top-Down Inference Neural Network

Our model is inspired by the recent success of RNNs [8, 25], which make use of dynamic sequential information in learning. RNNs are called *recurrent* model because they perform the same computation for every timestep, with the input dependent on the current inputs and previous outputs. We apply a similar idea to our layered label prediction problem: we consider each concept layer as an individual timestep, and model the label relations within and across concept layers in the recurrent learning framework.

Specifically, at each timestep t, we compute an image I^i's activations $a_t^i \in \mathbb{R}^{n_t \times 1}$ based on two terms: $a_{t-1}^i \in \mathbb{R}^{n_{t-1} \times 1}$, which are the activations from the last timestep $t-1$, and $x_t^i \in \mathbb{R}^{n_t \times 1}$, which are the activation from Eq. (1). The message passing process is defined as:

$$a_t^i = V_{t-1,t} \cdot a_{t-1}^i + H_t \cdot x_t^i + b_t, \tag{2}$$

where $V_{t-1,t} \in \mathbb{R}^{n_t \times n_{t-1}}$ are the inter-layer model parameters capturing the label relations between two consecutive concept layers in top-down order, $H_t \in \mathbb{R}^{n_t \times n_t}$ are the intra-layer model parameters to account for the label relations within each concept layer, and $b_t \in \mathbb{R}^{n_t \times 1}$ are the model biases. A sigmoid function can be applied on a_t^i to obtain label-specific prediction probabilities for image I^i.

Note that the inference process in Eq. (2) is different from the standard RNN learning: Eq. (2) unties $V_{t-1,t}$ and H_t in each timestep, while the standard RNNs learn the same V and H parameters over and over on all timesteps.

To learn the model parameters V's and H's, we apply a sigmoid function function σ on the activations a_t^i, and minimize the logistic cross-entropy loss with respect to V's and H's:

$$\begin{aligned} E(\{a_t^i\}) \quad = \quad & \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{y=1}^{n_t} \Big(\mathbb{1}(y_t^i = y) \cdot \log\left(\sigma(a_t^i)\right) \\ + \quad & \mathbb{1}(y_t^i \neq y) \cdot \log\left(1 - \sigma(a_t^i)\right) \Big), \end{aligned} \tag{3}$$

where $\mathbb{1}(z)$ is an indicator function which returns 1 if z is true and 0 otherwise.

3.1.2 BINN: Bidirectional Inference Neural Network

It makes more sense to model bidirectional inferences, as a concept layer is related to the two connected layers equally well. Therefore, we adopt the idea of bidirectional recurrent neural network [26], and propose the following bidi-

rectional inference model:

$$\overrightarrow{a}_t^i = \overrightarrow{V}_{t-1,t} \cdot \overrightarrow{a}_{t-1}^i + \overrightarrow{H}_t \cdot x_t^i + \overrightarrow{b}_t, \quad (4)$$

$$\overleftarrow{a}_t^i = \overleftarrow{V}_{t+1,t} \cdot \overleftarrow{a}_{t+1}^i + \overleftarrow{H}_t \cdot x_t^i + \overleftarrow{b}_t, \quad (5)$$

$$a_t^i = \overrightarrow{U}_t \cdot \overrightarrow{a}_t^i + \overleftarrow{U}_t \cdot \overleftarrow{a}_t^i + b_t. \quad (6)$$

where Eqs. (4) and (5) proceed top-down propagation and bottom-up propagation, respectively, and Eq. (6) aggregates the top-down and bottom-up messages into final activations for label prediction. Here $\overrightarrow{U}_t \in \mathbb{R}^{n_t \times n_t}$ and $\overleftarrow{U}_t \in \mathbb{R}^{n_t \times n_t}$ are aggregation model parameters, and we use the arrows $\rightarrow$ and $\leftarrow$ to indicate the directions of label propagation

As in the top-down inference model, the bidirectional inference model captures both inter-layer and intra-layer label relations in the model parameters V's and H's. For inter-layer relations, we connect a label in one concept layer to any label in its neighboring concept layers. For intra-layer relations, we model fully-connected relations within each concept layer. The model parameters V's, H's and U's are learned by minimizing the cross-entropy loss defined in Eq. (3).

3.1.3 SINN: Structured Inference Neural Network

The fully connected bidirectional model is capable of representing all types of label relations. In practice, however, it suffers from over-fitting due to limited training data and noise. To avoid this problem, we use a structured label relation graph to regularize information propagation.

We use structured label relations of positive correlation and negative correlation as prior knowledge to refine the model. Here is the intuition: since we know that *office* is an *indoor* scene, *beach* is an *outdoor* scene, and *indoor* and *outdoor* are mutually exclusive, a high score on *indoor* should increase the probability of label *office* and decrease the probability of label *beach*. Labels that are not semantically related, e.g. motorcycle and shoebox, should not effect each other. The structured label relations can be obtained from semantic taxonomies, or by parsing WordNet relations [22]. We describe the details of extracting label relations in Section 4.

We introduce the notation V^+, V^-, H^+ and H^- to explicitly capture structured label relations in between and within concept layers, where the superscripts $+$ and $-$ indicate positive and negative correlation, respectively. These model parameters are masked metrics capturing the label relations. Instead of learning full parametrized metrics of V^+, V^-, H^+ and H^-, we freeze some elements to be zero if there is no semantic relation between the corresponding labels. For example, V^+ models the positive correlation in between two concept layers: only the label pairs that have positive correlation have learnable model parameters, while the rest are zeroed out to remove potential noise. A similar setting goes to V^-, H^+ and H^-. Figure 3 shows an example positive correlation graph and a negative graph between two layers.

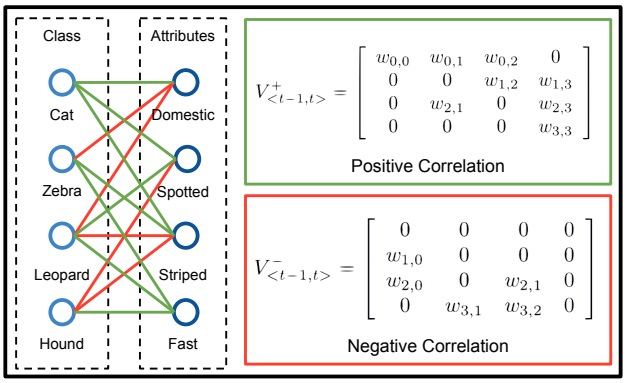

Figure 3. An example showing the model parameters V^+ and V^- between the *animal* layer and the *attribute* layer. Green edges in the graph represent positive correlation, and red edges represent negative correlation.

To implement the positive and negative label correlation, we propose the following structured message passing process:

$$\overrightarrow{a}_t^i = \gamma(\overrightarrow{V}^+_{t-1,t} \cdot \overrightarrow{a}_{t-1}^i) + \gamma(\overrightarrow{H}^+_t \cdot x_t^i) - \gamma(\overrightarrow{V}^-_{t-1,t} \cdot \overrightarrow{a}_{t-1}^i) - \gamma(\overrightarrow{H}^-_t \cdot x_t^i) + \overrightarrow{b}_t, \quad (7)$$

$$\overleftarrow{a}_t^i = \gamma(\overleftarrow{V}^+_{t+1,t} \cdot \overleftarrow{a}_{t+1}^i) + \gamma(\overleftarrow{H}^+_t \cdot x_t^i) - \gamma(\overleftarrow{V}^-_{t+1,t} \cdot \overleftarrow{a}_{t+1}^i) - \gamma(\overleftarrow{H}^-_t \cdot x_t^i) + \overleftarrow{b}_t, \quad (8)$$

$$a_t^i = \overrightarrow{U}_t \cdot \overrightarrow{a}_t^i + \overleftarrow{U}_t \cdot \overleftarrow{a}_t^i + b_t. \quad (9)$$

Here $\gamma(\cdot)$ stands for a ReLU activation function. It is essential for SINN as it enforces that activations from positive correlation always make positive contribution to output activation and keeps activations from negative correlation as negative contribution (notice the minus signs in Eqs (7) and (8)). To learn the model parameters V's, H's, and U's, we optimize the cross-entropy loss in Eq. (3).

3.2. Label Prediction

Now we introduce the method of predicting labels in test images with our model. As the model is trained with multiple concept layers, it is straightforward to recognize a label at each concept layer for the provided test image. This mechanism is called label prediction *without observation* (the default pipeline shown in Figure 2).

A more interesting application is to make predictions with *partial observations* – we want to predict labels in one concept layer given labels in another concept layers. Figure 4 illustrates the idea. Given an image shown in the left

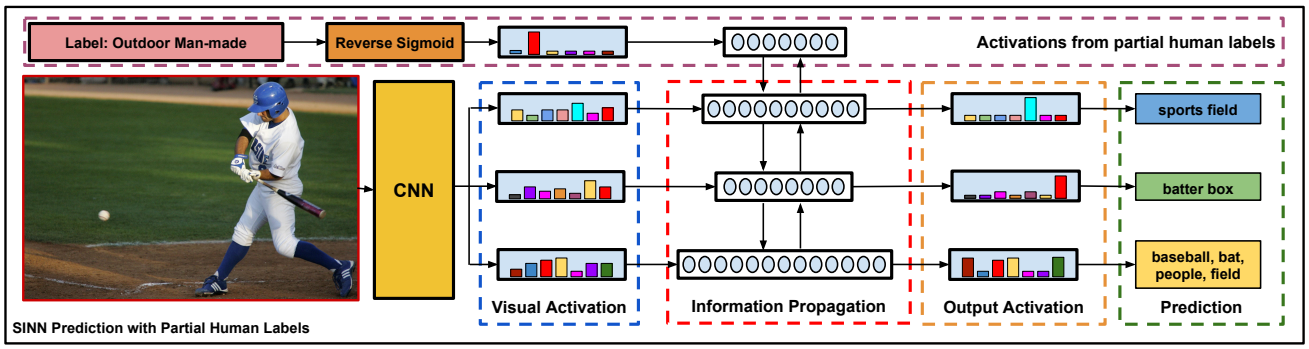

Figure 4. The label prediction pipeline with partial observation. The pipeline is similar to Figure 2 except that we now have a partial observation that this image is *outdoor man-made*. The SINN is able to take the observed label into consideration and improve the label predictions in the other concept layers.

side of Figure 4, we have more confidence to predict it as *batter box* once we know it is an *outdoor* image with attribute *sports field*.

To make use of the partially observed labels in our SINN framework, we need to transform the observed binary labels into soft activation scores for SINN to improve the label prediction on the target concept layers. Recall that SINN minimizes cross-entropy loss which applies sigmoid functions on activations to generate label confidences. Thus, we reverse this process by applying the inverse sigmoid function on the binary ground-truth labels to obtain activations. Formally, we define the activation a obtained from a ground-truth label y as:

$$a(y) = \log \frac{1}{1 - g(y)}, \tag{10}$$

$$g(y) = \begin{cases} y + \epsilon, \text{if } y = 0, \\ y - \epsilon, \text{if } y = 1. \end{cases} \tag{11}$$

Note that we put a small perturbation ϵ on the ground-truth label y for numerical stability. In our experiments, we keep $\epsilon = 0.001$.

3.3. Implementation Details

To optimize our learning objective, we use stochastic gradient descent with mini-batch size of 50 images and momentum of 0.9. For all training runs, we apply a two-stage policy as follows. In the first stage, we fixed pre-trained CNN networks, and train our SINN with a learning rate of 0.01 with fixed-size decay step. In the second stage, we set the learning rate as 0.0001 and fine-tune the CNN together with our SINN. We set the gradient clipping threshold to be 25 to prevent gradient explosion. The weight decay value for our training procedure is set to 0.0005.

In the computation of visual activations from the CNN, as different experiment datasets describe different semantic domains, we adopt different pretrained CNN models: ImageNet pretrained model [12] for experiments 4.1 and 4.2, placenet pretrained model [37] for experiment 4.3.

4. Experiments

We tested our method on three large-scale benchmark image datasets, the Animals with Attributes dataset (AwA) [19], the NUS-WIDE dataset [1], and the SUN397 dataset [36]. Each dataset has different concept layers and label relation graphs. Experimental results show that (1) our method effectively boosts classification performance using the label relation graphs; (2) our SINN model consistently outperforms baseline classifiers and related methods in all experiments; and (3) particularly, the SINN model achieves significant performance gain with partial human labels.

Dataset and Label relation generation The AwA dataset contains an 85-attribute layer, a 50-animal-category layer and a 28-taxonomy-term layer. We extract the label relations from the WordNet taxonomy knowledge graph [7, 10, 11]. The NUS-WIDE dataset is composed of Flickr images with 81 object category labels, 698 image group labels from image metadata, and 1000 noisy tags collected from users. We parse WordNet to obtain label similarity, and threshold the soft similarity values into positive and negative correlation for the label graph. The SUN397 label space has a typical hierarchical structure, with 397 fine-grained scene categories in the bottom layer, 16 general scene categories in the upper layer, and 3 coarsest categories in the top layer. The label relations are also extracted from WordNet.

Baseline. For each experiment, we compare our full method (CNN + SINN) with the baseline method: CNN + logistic regression. With further specifications, we may have extra baseline methods, such as CNN + BINN, CNN + logistic regression + extra tags, etc. We also compare our method with related state-of-the-art methods.

Evaluation metrics. We measure classification perfor-

mance by mean average precision (mAP) in all comparisons. mAP is a widely used metric for label-based retrieval and ranking. It measures the averaged performance over all label categories. In addition to mAP, we also adopted various metrics for special cases.

In the case of NUS-WIDE, the task is multi-label classification. We adopt the setting of [13] and report mAP per label (mAP_L) and mAP per image (mAP_I) for easy comparison. For comparison with related works ([21, 6, 13]) on NUS-WIDE, we also compute the per image and per label precisions and recalls. We abbreviate these metrics as $Prec_L$ for precision per label, $Prec_I$ for precision per image, Rec_L for recall per label, and Rec_I for precision per image.

For AwA and SUN397, we also compute the multi-class accuracy ($MCAcc$) and the intersection-over-union accuracy ($IoUAcc$). $MCAcc$ is a standard measurement for image classification problems. It averages per class accuracies as the final result. $IoUAcc$ is a common prediction measurement for multi-label classification, based on the hamming distance of predicted labels to ground-truth labels.

4.1. AwA: Layered Prediction with Label Relations

This experiment demonstrates the label prediction capability of our SINN model and the effectiveness of adding structured label relations for label prediction. We run each method five times with five random splits – 60% for training and 40% for test. We report the average performance as well as the standard deviation of each performance measure.

Note that there is very little related work with layered label prediction on AwA. The most relevant one is work by Hwang and Sigal [11] on unified semantic embedding (USE). The comparison is not strictly fair, as the train/test splits are different. Further, we include our BINN model without specifying the label relation graphs (see Section 3.1.2) as a baseline method in this experiment, as it can verify the performance gain in our model from including structure. The results are in Table 1.

Results. Table 1 shows that our method outperforms the baseline methods (CNN + Logistics and CNN + BINN variants) as well as the USE method, in terms of each concept layer and each performance metric. It validates the efficacy of our proposed model for image classification. Note that for the results in Table 1, we did not fine-tune the first seven layers of CNN [17] for fairer comparison with Hwang and Sigal [11] (which only makes use of DECAF features [5]). Fine-tuning the first seven layers of CNN could further improve $IoUAcc$ at each concept layer to 88.22 ± 0.38 (28 taxonomy terms), 69.17 ± 1.00 (50 animal classes), 86.06 ± 0.72 (85 attributes), and mAP_L to 96.72 ± 0.20 (28 taxonomy terms), 83.12 ± 0.69 (50 animal classes), 94.17 ± 0.55 (85 attributes), respectively.

4.2. NUS-WIDE: Multi-label Classification with Partial Human Labels of Tags and Groups

This experiment shows our model's capability to use noisy tags and structured tag-label relation to improve multi-label classification. The original NUS-WIDE dataset consists of 269,648 images collected from Flicker with 81 ground-truth concepts. As previous work used various evaluation metrics and experiment settings, and there are no fixed train/test splits, it is hard to make direct comparisons. And due to the lack of maintenance of this dataset, a faction of previously used images are unavailable now.

In order to make our result as comparable as possible, we tried to setup the experiments according to the previous work. We collected all available images and discard images with missing labels as previous work did [13, 6], and got 168,240 images of the original dataset. To make our result comparable with [13], we use 5 random splits with the same train/test ratio as [13] – there are 132,575 training images and 35,665 test images in each split.

To compare our method with [21, 13], we also used the tags and metadata groups in our experiment. Different from their settings, instead of augmenting images with 5000 tags, we only used 1000 tags, and augment the image with 698 group labels obtained from image medatada to form a three-layer group-concept-tag graph. Instead of using the tags as sparse binary input features (as in [21, 13]), we convert them to observed labels and feed them to our model.

Concept Layer	Method	$MCAcc$	$IoUAcc$	mAP_L
28 taxonomy terms	CNN + Logistics	-	80.41 ± 0.09	90.16 ± 0.10
	CNN + BINN	-	79.85 ± 0.13	89.92 ± 0.07
	CNN + SINN	-	$\mathbf{84.47 \pm 0.38}$	$\mathbf{93.00 \pm 0.29}$
50 animal classes	USE [11] + DECAF [5]	46.42 ± 1.33	-	-
	CNN + Logistics	78.44 ± 0.27	62.75 ± 0.26	78.35 ± 0.19
	CNN + BINN	79.00 ± 0.43	62.80 ± 0.25	78.88 ± 0.35
	CNN + SINN	$\mathbf{79.36 \pm 0.43}$	$\mathbf{66.60 \pm 0.43}$	$\mathbf{81.19 \pm 0.14}$
85 attributes	CNN + Logistics	-	81.29 ± 0.10	93.29 ± 0.12
	CNN + BINN	-	80.64 ± 0.13	93.04 ± 0.13
	CNN + SINN	-	$\mathbf{86.92 \pm 0.18}$	$\mathbf{96.05 \pm 0.07}$

Table 1. Layered label prediction results on the AwA dataset.

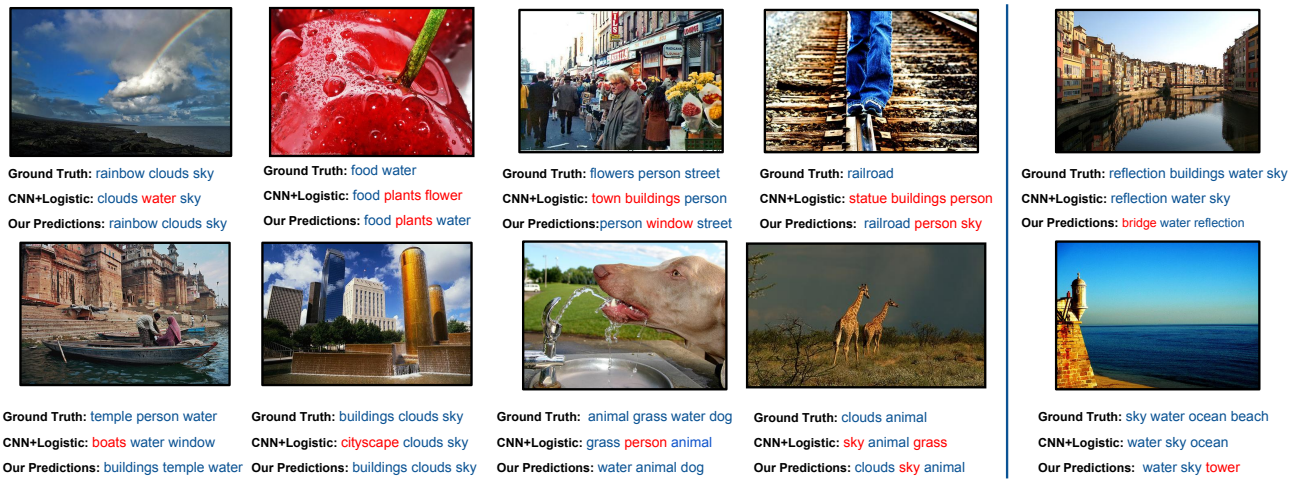

Figure 5. Visualization results (best viewed in color). We pick up 10 representative images from NUS-WIDE, and visualize the predicted labels of our method compared with CNN + Logistics. Under each image, we provide the ground-truth labels for ease of reference, and list the top-3 highest scoring predicted labels for each compared method. Correct predictions are marked in blue and incorrect predictions are in red. Failure cases are shown in the rightmost column.

The baselines for comparison are as follows. As our usual baseline, we extract features from a CNN pretrained on ImageNet [24] and train a logistic classifier on top of it. In addition, we set up a group of baselines that make use of the groups and tags as binary indicator feature vectors for logistic regression. These baselines serve as the control group to evaluate the quality of metadata we used in SINN. Next, a stronger baseline that uses both CNN output and metadata vector with logistic classifier was evaluated. This method has a similar setting as that of the state-of-art method by Johnson *et al*. [13], with difference in visual feature (CNN on image in our method versus CNN on image neighborhood) and tag feature (1k tag vector versus 5k tag vector).

We report our results on this dataset with two settings for our SINN, the first using 1k tags as the only observations to a bottom level of the relation graph. This method provides a good comparison to the tag neighborhood + tag vector [13], as we did not use extra information other than tags. In the second setting, we make both group and tag levels observable to our SINN, which achieves the best performance. We also compared our results with that of McAuley *et al*. [21], Gong *et al*. [6]. The results are summarized in Table 2. Note that we did not report our performance with fine-tuning the first seven layers of the CNN in this table, so as to make direct comparison of structured inference on SINN with our baseline method CNN + Logistics. Fine-tuned CNN with SINN improves mAP_L to 70.01 ± 0.40 and mAP_I to 83.68 ± 0.13.

Results. Table 2 shows that our proposed method outperforms all baseline methods and existing approaches (*e.g*., [13, 6, 21]) by a large margin. Note that the results are not directly comparable due to different settings in train/test splits. However, the results show that, by modeling label relations between tags, groups and concepts, our model achieves dramatic improvement on visual prediction.

Method	mAP_L	mAP_I	Rec_L	$Prec_L$	Rec_I	$Prec_I$
Graphical Model [21]	49.00	-	-	-	-	-
CNN + WARP [6]	-	-	35.60	31.65	60.49	48.59
5k tags + Logistics [13]	43.88 ± 0.32	77.06 ± 0.14	47.52 ± 2.59	46.83 ± 0.89	71.34 ± 0.16	51.18 ± 0.16
Tag neighbors + 5k tags [13]	61.88 ± 0.36	80.27 ± 0.08	57.30 ± 0.44	54.74 ± 0.63	75.10 ± 0.20	53.46 ± 0.09
CNN + Logistics	46.94 ± 0.47	72.25 ± 0.19	45.03 ± 0.44	45.60 ± 0.35	70.77 ± 0.21	51.32 ± 0.14
1k tags + Logistics	50.33 ± 0.37	66.57 ± 0.12	23.97 ± 0.23	47.40 ± 0.07	64.95 ± 0.18	47.40 ± 0.07
1k tags + Groups + Logistics	52.81 ± 0.40	68.04 ± 0.12	25.54 ± 0.24	49.26 ± 0.15	65.99 ± 0.15	48.13 ± 0.05
1k tags + Groups + CNN + Logistics	54.67 ± 0.57	77.81 ± 0.22	50.83 ± 0.53	49.36 ± 0.30	75.38 ± 0.16	54.61 ± 0.09
1k tags + CNN + SINN	67.20 ± 0.60	81.99 ± 0.14	59.82 ± 0.12	57.02 ± 0.57	78.78 ± 0.13	56.84 ± 0.07
1k tags + Groups + CNN + SINN	$\mathbf{69.24 \pm 0.47}$	$\mathbf{82.53 \pm 0.15}$	$\mathbf{60.63 \pm 0.67}$	$\mathbf{58.30 \pm 0.33}$	$\mathbf{79.12 \pm 0.18}$	$\mathbf{57.05 \pm 0.09}$

Table 2. Results on NUS-WIDE. We measure precision Pre_L, Pre_I and recall Rec_L, Rec_I with n = 3 labels for each image.

Concept Layer	Method	$MCAcc$	$IoUAcc$	mAP_L
3 coarse scene categories	CNN + Logistics	-	83.67 ± 0.18	95.19 ± 0.07
	CNN + BINN	-	83.63 ± 0.24	95.19 ± 0.03
	CNN + SINN	-	$\mathbf{85.95 \pm 0.44}$	$\mathbf{96.40 \pm 0.18}$
16 general scene categories	CNN + Logistics	-	64.30 ± 0.27	83.30 ± 0.19
	CNN + BINN	-	63.40 ± 0.35	82.93 ± 0.14
	CNN + SINN	-	$\mathbf{66.46 \pm 1.10}$	$\mathbf{84.97 \pm 0.96}$
397 fine-grained scene categories	Image features + SVM [36, 35]	42.70	-	-
	CNN + Logistics	$\mathbf{57.86 \pm 0.38}$	35.97 ± 0.37	55.31 ± 0.30
	CNN + BINN	57.52 ± 0.29	35.44 ± 1.02	55.57 ± 0.63
	CNN + SINN	57.60 ± 0.38	$\mathbf{37.71 \pm 1.13}$	$\mathbf{58.00 \pm 0.33}$

Table 3. Layered label prediction results on the SUN397 dataset.

We visualize some results in Figure 5 showing exemplars on which our method improves over baseline predictions.

4.3. SUN397: Improving Scene Recognition with and without partially Observed Labels

We conducted two experiments on the SUN397 dataset. The first experiment is similar to the study on AwA: we applied our model to layered image classification with label relations, and compare our model with CNN + Logistics and CNN + BINN baselines, as well as a state-of-the-art approach [36, 35]. For fair comparison, we used the same train/test split ratio as [36, 35], where we have 50 training and test images in each of the 397 scene categories. To migrate the randomness in sampling, we also repeat the experiment 5 times and report the average performance as well as the standard deviations. The results are summarized in Table 3, showing that our proposed method again achieves a considerable performance gain over all the compared methods.

In the second experiment, we considered partially observed labels from the top (coarsest) scene layer as input to our inference framework. In other words, we assume we know whether an image is *indoor*, *outdoor man-made*, or *outdoor natural*. We compare the 397 fine-grained scene recognition performance in Table 4. We compare to a set of baselines, including CNN + Logistics + Partial Labels, that considers the partial labels as an extra binary indicator feature vector for logistic regression. Results show that our method combined with partial labels (*i.e.*, CNN + SINN + Partial Labels) improves over baselines, exceeding the second best by 4% $MCAcc$ and 6% mAP_L.

Method	$MCAcc$	mAP_L
Image features + SVM [36, 35]	42.70	-
CNN + Logistics	57.86 ± 0.38	55.31 ± 0.30
CNN + BINN	57.52 ± 0.29	55.57 ± 0.63
CNN + SINN	57.60 ± 0.38	58.00 ± 0.33
CNN + Logistics + Partial Labels	59.08 ± 0.27	56.88 ± 0.29
CNN + SINN + Partial Labels	$\mathbf{63.46 \pm 0.18}$	$\mathbf{64.63 \pm 0.28}$

Table 4. Recognition results on the 397 fine-grained scene categories. Note that the last two compared methods make use of partially observed labels from the top (coarsest) scene layer, *i.e.*, *indoor*, *outdoor man-made*, and *outdoor natural*.

5. Conclusion

We have presented a structured inference neural network (SINN) for layered label prediction. Our model makes use of label relation graphs and concept layers to argument inference of semantic image labels. Beyond this, our model can be flexibly extended to consider partially observed human labels. We borrow the idea of RNN to implement our SINN model, and combine it organically with an underlying CNN visual output. Experiments on three benchmark image datasets show the effectiveness of the proposed method in standard image classification tasks. Moreover, we also demonstrate empirically that label prediction is further improved once partially observed human labels are fed into the SINN.

References

[1] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *CIVR*, 2009. 1, 2, 5

[2] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*. 2014. 1, 2

[3] J. Deng, S. Satheesh, A. C. Berg, and F. Li. Fast and balanced: Efficient label tree learning for large scale object recognition. In *NIPS*, 2011. 2

[4] N. Ding, J. Deng, K. Murphy, and H. Neven. Probabilistic label relation graphs with ising models. *ICCV*, 2015. 1, 2

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *ICML*, 2013. 6

[6] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *ICLR*, 2014. 6, 7

[7] K. Grauman, F. Sha, and S. J. Hwang. Learning a tree of metrics with disjoint visual features. In *NIPS*, 2011. 2, 5

[8] A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In *NIPS*, 2008. 3

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation (NC)*, 9(8):1735–1780, 1997. 2

[10] S. J. Hwang, K. Grauman, and F. Sha. Semantic kernel forests from multiple taxonomies. In *NIPS*, 2012. 2, 5

[11] S. J. Hwang and L. Sigal. A unified semantic embedding: Relating taxonomies and attributes. In *NIPS*, 2014. 5, 6

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014. 5

[13] J. Johnson, L. Ballan, and F.-F. Li. Love thy neighbors: Image annotation by exploiting image metadata. *ICCV*, 2015. 2, 6, 7

[14] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *CVPR*, 2015. 2

[15] P. Kohli, J. Rihan, M. Bray, and P. H. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *International Journal of Computer Vision (IJCV)*, 79(3):285–298, 2008. 2

[16] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? text-to-image coreference. In *CVPR*, 2014. 2

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2, 3, 6

[18] M. Kumar, P. Ton, and A. Zisserman. Obj cut. In *CVPR*, 2005. 2

[19] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 36(3):453–465, 2014. 1, 2, 5

[20] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop*, 2004. 2

[21] J. McAuley and J. Leskovec. Image labeling on a network: using social-network metadata for image classification. In *ECCV*. 2012. 2, 6, 7

[22] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM (CACM)*, 38(11):39–41, 1995. 4

[23] V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. Berg. From large scale image categorization to entry-level categories. In *ICCV*, 2013. 2

[24] 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 (IJCV)*, pages 1–42, 2015. 1, 7

[25] H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, 2014. 3

[26] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing (TSP)*, 45(11):2673–2681, 1997. 2, 3

[27] A. G. Schwing and R. Urtasun. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351*, 2015. 2

[28] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2014. 1

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[30] 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. 1

[31] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, 2003. 2

[32] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 2

[33] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005. 2

[34] B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *CVPR*, 2007. 2

[35] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision (IJCV)*, pages 1–20, 2014. 8

[36] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, et al. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 2, 5, 8

[37] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 5