Exploit Bounding Box Annotations for Multi-label Object Recognition

Hao Yang[1], Joey Tianyi Zhou[2], Yu Zhang[3], Bin-Bin Gao[4], Jianxin Wu[4], and Jianfei Cai[1]

[1]SCE, Nanyang Technological University, lancelot365@gmail.com, ASJFCai@ntu.edu.sg
[2]IHPC, A*STAR, zhouty@ihpc.a-star.edu.sg
[3]Bioinformatics Institute, A*STAR, zhangyu@bii.a-star.edu.sg
[4]National Key Laboratory for Novel Software Technology, Nanjing University, China
gaobb@lamda.nju.edu.cn, wujx2001@nju.edu.cn

Abstract

Convolutional neural networks (CNNs) have shown great performance as general feature representations for object recognition applications. However, for multi-label images that contain multiple objects from different categories, scales and locations, global CNN features are not optimal. In this paper, we incorporate local information to enhance the feature discriminative power. In particular, we first extract object proposals from each image. With each image treated as a bag and object proposals extracted from it treated as instances, we transform the multi-label recognition problem into a multi-class multi-instance learning problem. Then, in addition to extracting the typical CNN feature representation from each proposal, we propose to make use of ground-truth bounding box annotations (strong labels) to add another level of local information by using nearest-neighbor relationships of local regions to form a multi-view pipeline. The proposed multi-view multi-instance framework utilizes both weak and strong labels effectively, and more importantly it has the generalization ability to even boost the performance of unseen categories by partial strong labels from other categories. Our framework is extensively compared with state-of-the-art hand-crafted feature based methods and CNN based methods on two multi-label benchmark datasets. The experimental results validate the discriminative power and the generalization ability of the proposed framework. With strong labels, our framework is able to achieve state-of-the-art results in both datasets.

1. Introduction

Recently, the availability of large amount of labeled data has greatly boosted the development of feature learning methods for classification. In particular, convolutional neural networks (CNNs) achieve great success in visual recognition/classification tasks. Features extracted from CNNs can provide powerful global representations for the single object recognition problem [15, 22, 23]. However, conventional CNN features may not generalize well for images containing multiple objects as the objects can be in different locations, scales, occlusions and categories. Fig. 1 shows an example of such images. Since multi-label recognition task is more general and practical in real world applications, many CNN related methods [18, 19, 26] have been proposed to address the problem.

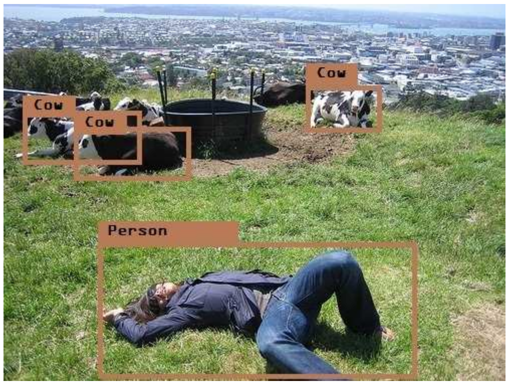

Figure 1. An example of a typical multi-label image, which contains several cows in different locations as well as a person.

A well known fact is that image-level labels can be utilized to fine-tune a pre-trained CNN model and produce good global representations [22, 23, 19]. However, due to the diversity and the complexity of multi-label images, classifiers trained from such global representations might not be optimal. For example, if we use images similar to Fig. 1 to train a classifier for "person", the classifier will have to account for not only hundreds of different variations of “person” but also other objects contained in the images. The complexity of multi-label images adds an extra level of difficulty for training appropriate classifiers with the global image representations. Furthermore, due to the large intra-class variations of multi-label images, the global features extracted from training images are likely to be unevenly distributed in the feature space. A classifier trained with such features can be successful at regions that are densely populated with training instances, but may fail in poorly sampled areas of the feature space [31].

To address the problems with global CNN representations, following the recent works [19, 26, 18, 12], we incorporate local information via extracting object proposals using general object detection techniques such as selective search [25] for multi-label object recognition. By decomposing an image into local regions that could potentially contain objects, we avoid the complex process of directly recognizing multiple objects in the whole image. Instead, we only need to identify whether there exist target objects in the local regions. However, as the local regions are noisy and of large variations (see Fig. 3), the usual CNN representations might not be good enough for discrimination. Therefore, we add another level of locality by incorporating local nearest-neighbor relationships of these local regions. In this way, the resulting features will be more evenly distributed in the feature space. However, such relationships are not easy to obtain only through weak supervision, i.e., image-level labels. Fortunately, for many multi-label applications, we can exploit the strong supervision information, i.e. ground-truth bounding boxes, which can be considered as local regions with strong labels. Then, we can exploit the relationships between object proposals and ground-truth bounding boxes (e.g., nearest neighbor relationships) to help multi-label recognition.

We would like to point out that ground-truth bounding boxes have been utilized in two proposal-based methods [18, 12] for multi-label object recognition. In particular, [12] makes use of ground-truth bounding boxes to train category-specific classifiers to classify object proposals. However, it requires ground-truth bounding boxes for each category of objects, which might not be available in practice. In contrast, for our proposed method, even with only partial strong labels (e.g., bounding boxes and labels for 10 classes in the 20 classes of Pascal VOC), the proposed local relationships generalize well and can help recognize all classes (e.g., improving recognition of the 20 classes in VOC). [18] directly uses ground-truth bounding boxes to fine tune the CNN model, but its performance is not better than other proposal-based methods [19, 26] that do not utilize ground-truth bounding boxes. In other words, an effective way to exploit ground-truth bounding boxes for multi-label object recognition is still missing, which is what we aim to provide in this paper.

Fig. 2 gives an overview of our proposed framework. We utilize both strong and weak labels as two *views*, and propose a multi-view multi-instance framework to tackle the multi-label object recognition task. In particular, for any image, we first extract object proposals using general object detection techniques. The global image and its accompanying weak (image) label is used to fine-tune a standard CNN to generate a *feature view* representation for each proposal. Using the ground truth bounding boxes and their strong labels, we design a large margin nearest neighbor (LMNN) CNN architecture to learn a low-dimensional feature so that we could extract nearest neighbor relationship between local regions and a candidate pool formed by ground truth objects. These local NN features are used as the *label view*. When combining both views, we can achieve a balance between global semantic abstraction and local similarity, hence enhancing the discriminative power of our framework. More importantly, as the strong labels are indirectly utilized through LMNN to encode local neighborhood relationships among labelled local regions, the proposed framework can generalize well to the whole local region space, even with only partial strong labels for part of the object classes, making our framework more practical.

The main contribution of this research lies in the proposed multi-view multi-instance framework, which utilizes bounding box annotations (strong labels) to encode the label view and combine it with the typical CNN feature representation (feature view) for multi-label object recognition. Another novelty of our work is the proposed LMNN CNN which effectively extracts local information from the strong labels.

2. Related Works

Our paper mainly relates to the topics of CNN based multi-label object recognition, multi-view and multi-instance learning and local and metric learning.

CNN based multi-label object recognition. Recently, CNN models have been adopted to solve the multi-label object recognition problem. Many works [12, 22, 18, 19, 26] have demonstrated that CNN models pre-trained on a large dataset such as ILSVRC can be used to extract features for other applications without enough training data. A typical way ([22], [23] and [5]) is to directly apply a pre-trained CNN model to extract an off-the-shelf global feature for each image from a multi-label dataset, and use these features for classification. However, different from single-object images from the ImageNet, multi-label images usually have multiple objects in different locations, scales and occlusions, and thus global representations are not optimal for solving the problem [26]. More recently, two proposal-based methods [18, 12] were propose for multi-label recognition and detection tasks with the help of ground truth bounding boxes. These methods achieve significant im-

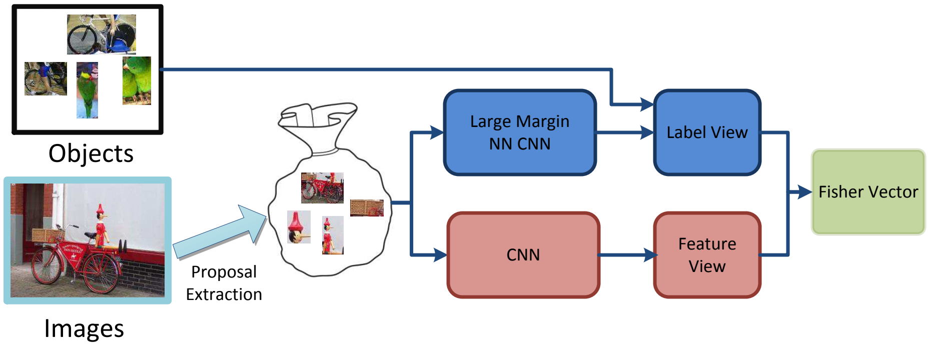

Figure 2. Overview of the proposed multi-view multi-instance framework. We transform the multi-label object recognition problem into a multi-class multi-instance learning problem by first extracting object proposals from each image using selective search. Two types of features are then extracted for each proposal. One is a low-dimensional feature from a large-margin nearest neighbor (LMNN) CNN, which is used to generate the label view by encoding the label information of k-NN from the candidate pool (containing ground truth objects). The other is a standard CNN feature as the feature view. These two views are fused and then used to encode a Fisher vector for each image.

provement over single global representations. On the other hand, [19] and [26] handle the problem in a weakly supervised manner by max-pooling image scores from the proposal scores. Moreover, [24] employs a very deep CNN, aggregates multiple features from different scales of the image and achieves state-of-the-art results.

Multi-view and multi-instance learning. Multi-view learning deals with data from multiple sources or feature sets. The goal of multi-view learning is to exploit the relationship between views to improve the performance or reduce model complexity. Multi-view learning is well studied in conjunction with semi-supervised learning or active learning. To combine information from multi-views for supervised learning, fusion techniques at feature level or classifier level can be employed [32]. Multi-instance learning aims at separating bags containing multiple instances. Over the years, many multi-instance learning algorithms have been proposed, including miBoosting [29], miSVM [1], MILES [7] and miGraph [33]. Several works also studied the combination of multi-view and multi-instance learning and its application to computer vision tasks.

Local and metric learning. Existing local learning methods mainly vary in the way that they utilize the labelled instances nearest to a test instance. One way is to only use a fixed number of nearest neighbors to the test point to train a model using neural network, SVM or just voting. The other way is to learn a transformation of the feature space (e.g., Linear Discriminant Analysis). In either way, the learned model can be better tailored for the test instance's neighborhood property [13]. Metric learning is closely related to local learning as a good distance metric is crucial for the success of local learning. Generally, metric learning methods optimize a distance metric to best satisfy known similarity constraints between training data [3]. Some metric learning methods learn a single global metric [27]. Others learn local metrics that vary in different regions of the feature space [30].

3. Multi-Label as Multi-Instance

In this section, we introduce the first level of locality by formulating the multi-label object recognition problem as a multi-instance learning (MIL) problem. To be specific, given a set of n training images $\{\boldsymbol{X}_i\}_{i=1}^n$, we extract n_i object proposals $\{\boldsymbol{x}_{ij}, j = 1, \ldots, n_i\}$ from each image $\boldsymbol{X}_i$ using general object detection techniques. By decomposing images into object proposals, each image $\boldsymbol{X}_i$ becomes a bag containing several positive instances, i.e., proposals with the target objects, and negative instances, i.e., proposals with background or other objects. The problem of classifying $\boldsymbol{X}_i$ is thus transformed from a multi-label classification problem to a multi-class MIL problem. The merit of such a transformation is that we do not need to deal with the complex process of directly recognizing multiple objects in multiple scales, locations and categories in a single image. Instead, we only need to identify whether there exist target objects in the proposals, which has been proven to be the forte of CNN features [12].

MIL problems assume that every positive bag contains at least one positive instance. As extensively compared and evaluated in [14], state-of-the-art general object detection methods like BING [8], selective search [25], MCG [2] and EdgeBoxes [34] can reach reasonably good recall rates with several hundreds of proposals. Therefore, if we sample enough proposals from each image, we can safely assume that these proposals can cover all objects (or at least all object categories) in an image, thus fulfilling the assumption of multi-instance learning.

In particular, we employ the unsupervised selective search method [25] for object proposal generation. Selective search has proven to be able to achieve a balance between effectiveness and efficiency [14]. More importantly, as it is unsupervised, no extra training data or ground truth

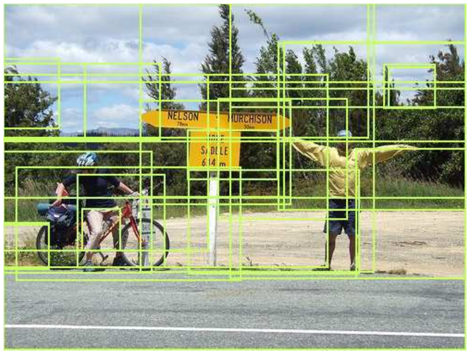

Figure 3. An example of object proposals generated by selective search. We demonstrate 30 randomly sampled proposals from the full 218 proposals, which clearly cover two of the main objects: person and bike.

bounding boxes are needed in this stage. Example of proposals extracted by selective search can be found in Fig. 3.

Traditionally, MIL is formulated as a max-margin classification problem with latent parameters optimized using alternating optimization. Typical examples include miSVM [1] and Latent SVM [11]. However, although these methods can achieve satisfactory accuracies, their limitations in scalability hinder their applicability to current large scale image classification tasks. For large scale MIL problem, [28] shows that Fisher vector [20, 21] (FV) can be used as an efficient and effective holistic representation for a bag. Thus, we choose to represent each bag $\boldsymbol{X}_i$ as an FV.

Assume we have a K-component Gaussian Mixture Model (GMM) with parameters $\theta = \{\omega_k, \boldsymbol{\mu}_k, \Sigma_k, k = 1, \ldots, K\}$, where ω_k, $\boldsymbol{\mu}_k$ and Σ_k are the mixture weight, mean vector and covariance matrix of the k-th Gaussian, respectively. The covariance matrices Σ_k are assumed to be diagonal, where the corresponding standard deviations of the diagonal entries form a vector $\boldsymbol{\sigma}_k$. We have [21]:

$$f_{\boldsymbol{\mu}_k}^{\boldsymbol{X}_i} = \frac{1}{\sqrt{\omega_k}} \sum_{j=1}^{n_i} \gamma_j(k) \left(\frac{\boldsymbol{x}_{ij} - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right), \tag{1}$$

$$f_{\sigma_k}^{\boldsymbol{X}_i} = \frac{1}{\sqrt{\omega_k}} \sum_{j=1}^{n_i} \gamma_j(k) \frac{1}{\sqrt{2}} \left[\frac{(\boldsymbol{x}_{ij} - \boldsymbol{\mu}_k)^2}{\boldsymbol{\sigma}_k^2} - 1 \right], \tag{2}$$

where $\gamma_j(k)$ is the soft assignment weight, which is also the probability for x_{ij} to be generated by the k-th Gaussian:

$$\gamma_j(k) = p(k|x_{ij}, \theta). \tag{3}$$

We map all the proposals $\{\boldsymbol{x}_{ij}, j = 1, \ldots, n_i\}$ in an image $\boldsymbol{X}_i$ to an FV by concatenating $f_{\boldsymbol{\mu}_k}^{\boldsymbol{X}_i}$ and $f_{\boldsymbol{\sigma}_k}^{\boldsymbol{X}_i}$ for all $k = 1, \ldots, K$, and denoting it as $F^{\boldsymbol{X}_i}$. $F^{\boldsymbol{X}_i}$ will be used as the final feature to train the one-vs-all linear classifiers. Note that for simplicity, we abuse the notation of $\boldsymbol{x}_{ij}$ for both proposal j in image i and its corresponding feature representation. In the next section, we will describe how to generate the feature representation $\boldsymbol{x}_{ij}$ for each proposal.

4. From Global Representation to Local Similarity

Once we obtain object proposals for each image, we can naturally use CNN features to represent these proposals. Following general practices in the literature [22, 23], each proposal is fed into a pre-trained CNN, and the output of the second last fully connected layer (e.g. layer 7 in AlexNet [15]) is used as the feature representation of that particular proposal. We call this kind of representation as the feature view $\boldsymbol{f}_{ij}$ for proposal $\boldsymbol{x}_{ij}$ from image $\boldsymbol{X}_i$. With the proposals represented by CNN features, one baseline is to encode each image (bag) at the feature view by Fisher vector as discussed in Sec. 3. Such baseline is able to get reasonably good results by utilizing the Fisher vector generated only from the feature view.

However, since the proposals contain different objects as well as random background, there exist large variances and imbalanced distributions. As a consequence, the global representation might not be accurate enough. Inspired by the idea of local learning, which solves the data density and intra-class variation problems by focusing on a subset of the data that are more relevant to a particular instance [4], we propose the second level of locality by adding local spatial configuration information as the label view (cf. Fig. 2) to enhance the discriminative power of the feature.

To effectively encode the local spatial configuration of a proposal, we need to solve two key problems: how to form a good candidate pool for local learning and how to determine which candidates are relevant to a particular new proposal. For the former problem, since we have some ground truth object bounding boxes from the strong labels, we could use them as the candidate pool assuming all of the ground truth objects are useful. For the latter one, we follow the common assumption that the most relevant candidates are the nearest neighbors. In this way, the problem becomes how to define "nearest". Many studies [3, 27] have shown that the distance metric is critical to the performances of local learning.

4.1. CNN as Metric Learning

Metric learning studies the problem of learning a discriminative distance metric. Conventional Mahalanobis metric learning methods optimize the parameters of a distance function in order to best satisfy known similarity or dissimilarity constraints between training instances [3]. To be specific, given a set of n labelled training instances $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, the goal of metric learning is to learn a square matrix $\mathbf{M}$ such that the distance mapped between training data, represented as

$$D_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \mathbf{M} (\boldsymbol{x}_i - \boldsymbol{x}_j)\,, \tag{4}$$

satisfies certain constraints. Since $\mathbf{M}$ is symmetric and positive semi-definite, it can be decomposed as $\mathbf{M} = \mathbf{W}^T\mathbf{W}$, and $D_M(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be rewritten as:

$$D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \|\mathbf{W}(\boldsymbol{x}_i - \boldsymbol{x}_j)\|^2\,. \tag{5}$$

We can see that learning a distance metric is equivalent to learning linear projection W that maps the data from input space to a transformed space. In this sense, the extraction of CNN features from the original raw pixel space can also be viewed as a form of metric learning, while only the process is highly nonlinear. However, the goal of CNN is usually to minimize the classification error using loss functions such as the logistic loss, which may not be suitable for local encoding.

Our desired metric should be discriminative such that all categories are well separated, as well as compact so that we can find more accurate nearest neighbors. Specifically, we want the pairwise distance between instances from the same class to be smaller than that between instances from different classes. In order to achieve such a goal, [27] proposed the large-margin distance to minimize the following objective function:

$$\begin{aligned} &\sum_{i,j} \eta_{ij} D(\boldsymbol{x}_i, \boldsymbol{x}_j) + \\ \alpha &\sum_{i,j,l} \eta_{ij}(1 - y_{il})\left[1 + D(\boldsymbol{x}_i, \boldsymbol{x}_j) - D(\boldsymbol{x}_i, \boldsymbol{x}_l)\right]_+\,. \end{aligned} \tag{6}$$

Here η encodes target nearest neighbor information, where $\eta_{ij} = 1$ if $\boldsymbol{x}_j$ is one of the $\hat{k}$ positive nearest neighbors of $\boldsymbol{x}_i$; otherwise $\eta_{ij} = 0$. y is the label information where $y_{il} = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$ are in the same class; otherwise $y_{il} = 0$. $[\cdot]_+ = \max(\cdot, 0)$ is the hinge loss function. α is the trade-off parameter. The first term in Eq. 6 penalizes large distances between instances and target neighbors, and the second term penalizes small distances between each instance and all other instances that do not share the same label. By employing such an objective function, we can ensure that the $\hat{k}$-nearest neighbors of an instance belong to the same class, while instances from different classes are separated by a large margin.

In order to learn a discriminative metric, we propose to learn a large-margin nearest neighbor (LMNN) CNN. Specifically, we replace the logistic loss with the large margin nearest neighbor loss and train a network with low-dimension output utilizing the strong labels. Details of training and fine-tuning the LMNN CNN can be found in Section 4.3. The output of the proposed LMNN network is a low-dimensional feature that shares the good semantic abstraction power of conventional CNN feature and the good neighborhood property of large-margin metric learning. We then build the candidate pool with the LMNN CNN features extracted from ground truth objects.

4.2. Encoding Local Spatial Label Distribution as Label View

To effectively incorporate local label information around a local region, we encode its neighborhood as the label view. Specifically, we extract features from each proposal $\boldsymbol{x}_{ij}$ using the LMNN CNN, then find k nearest neighbors of $\boldsymbol{x}_{ij}$ in the candidate feature pool as $\boldsymbol{nn}_{ij} = \{nn_{ij}^1, \dots, nn_{ij}^k\}$ and record their labels $\boldsymbol{l}_{ij} = \left[l_{ij}^1 \; \dots \; l_{ij}^k\right]$. The label information (e.g. l_{ij}^k) of a neighbor (e.g. nn_{ij}^k) is encoded as a C-dimensional binary vector, which corresponds to C categories. The d-th dimension $l_{ij}^k(d) = 1$, $(d = 1, \dots, C)$ if the object is annotated as class d; otherwise $l_{ij}^k(d) = 0$. Therefore, $\boldsymbol{l}_{ij}$ is a $1 \times kC$ vector and it will be used as the feature for the label view.

The merit of such indirectly utilizing ground truth bounding boxes as the label view is the good generalization ability. As the label view is a form of local structure representation, even for unseen categories, i.e. no same-category strong labels, the encoding process can naturally exploit existing semantically or visually close categories to build local support. For example, suppose we do not have the bounding box annotations for “cat” and “train”, a proposal containing “cat” might have nearest neighbors of “dog”, “tiger” or other related animals, and a proposal containing “train” might have nearest neighbors of “car”, “truck” or other vehicles. Although lacking the exact annotations of certain category, the label view is still able to encode the local structure with semantically or visually similar objects. In this way, our framework can make use of existing strong supervision information to boost the overall performance. The experimental results shown in Section 5.3 validate this argument.

We directly concatenate the feature view and the label view to form the final representation of each proposal $\boldsymbol{x}_{ij}$ as $\left[\boldsymbol{f}_{ij} \; \lambda \boldsymbol{l}_{ij}\right]$, where λ is the trade-off parameter between the feature view and the label view.

4.3. Network Configurations and Implementation Details

Our framework consists of two networks, a large-margin nearest neighbor (LMNN) CNN and a standard CNN. Both networks’ architectures are similar to [5] with 5 convolutional layers and 3 fully-connected layers, and the dimension of the layer-7 output is set to 2048. The main differences of these two networks lie in the loss function and the fine-tuning process. For LMNN CNN, its layer-8 output is a 128-dimensional feature, based on which we measure the pair-wise distance for kNN, while the output of the standard

CNN is a C-dimensional score vector, corresponding to the C categories.

Data pre-processing and pre-training. We use the ILSVRC 2012 dataset to pre-train both networks. Given an image, we resize the short side to 224 with bilinear interpolation and perform a center crop to generate the standard 224×224 input. Each of these inputs is then pre-processed by subtracting the mean of ILSVRC images.

Fine-tuning. To better adopt the pre-trained network for specific applications, we also fine-tune these networks using task relevant data. Unlike [5], currently our implementation does not involve any data augmentation in the fine-tuning stage.

For the standard CNN used for feature view, we only fine-tune the network with weak labels on the whole image. As our task is multi-label recognition, following [26], we use square loss instead of the logistic loss. To be specific, suppose we have a label vector $\boldsymbol{y}_i = [y_{i1}, y_{i2}, \dots, y_{iC}]$ for the i-th image. $y_{ij} = 1$ $(j = 1, \dots, C)$ if the image is annotated with class j; otherwise $y_{ij} = 0$. The ground truth probability vector of the i-th image is defined as $\boldsymbol{p}_i = \boldsymbol{y}_i / \|\boldsymbol{y}_i\|_1$ and the predicted probability vector is $\hat{\boldsymbol{p}}_i = [\hat{p}_{i1}, \hat{p}_{i2}, \dots, \hat{p}_{iC}]$. Then, the cost function to be minimized is defined as

$$\frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{C} (p_{ij} - \hat{p}_{ij})^2 \,. \tag{7}$$

During the fine-tuning, the parameters of the first seven layers of the network are initialized with the pre-trained parameters. The parameters of the last fully connected layer is initialized with a Gaussian distribution. We tune the network for 10 epochs in total.

For the large-margin NN (LMNN) CNN used for label view, we execute a three-step fine-tuning. The first step is the image level fine-tuning similar to the process we have described above. The second step is ground truth objects fine-tuning, where we fine-tune the network using ground truth objects with the logistic loss. The final step is the large-margin nearest neighbor fine-tuning, where we fine-tune the network with the loss function of Eq. 6 described in Section 4.1. To accelerate the process, in this final step, we fix all parameters of the first seven layers and only fine-tune the parameters of the last fully connected layer.

5. Experimental Results

In this section, we present the experimental results of the proposed multi-view multi-instance framework on multi-label object recognition tasks.

5.1. Datasets and Baselines

We evaluate our method on the PASCAL Visual Object Classes Challenge (VOC) datasets [10], which are widely used as benchmark datasets for the multi-label object recognition task. In particular, we use the VOC 2007 and VOC 2012 datasets. The details of these datasets can be found in Table 1. These two datasets have a pre-defined split of TRAIN, VAL and TEST sets. We use TRAIN and VAL for training and TEST for testing. The evaluation method is average precision (AP) and mean average precision (mAP).

Table 1. Dataset information.

Dataset	#TrainVal	#Test	#Classes
VOC 2007	5011	4952	20
VOC 2012	11540	10991	20

We compare the proposed framework with the following state-of-the-art approaches:

- CNN-SVM [23]. This method employed OverFeat [22], which is pre-trained on ImageNet, to get CNN activations as the off-the-shelf features. Specifically, CNN-SVM employs the 4096-d feature extracted from the 22-nd layer of OverFeat and uses these features to train a linear SVM for the classification task.
- PRE [18]. [18] proposed to transfer image representations learned with CNN on ImageNet to other visual recognition tasks with limited training data. The network has exactly the same architecture as that in [15]. The network is first pre-trained on ImageNet. The parameters of the first seven layers of CNN are then fixed and the last fully-connected layer is replaced by two adaptation layers. Finally, the adaptation layers are trained with images from the target dataset.
- HCP [26]. HCP proposed to solve the multi-label object recognition task by extracting object proposals from the images. Specifically, HCP has three main steps. The first step is to pre-train a CNN on ImageNet data. The second step is image-level fine-tuning that uses image labels and square loss to fine-tune the pre-trained CNN. The final step is to employ BING [8] to extract object proposals and fine-tune the network with these proposals. The image-level scores are obtained by max-pooling from the scores of the proposals.
- [19]. [19] also handled the problem in a weakly supervised manner. Particularly, multiple windows are extracted from different scales of the images in the dense sampling fashion. The scores of these windows are combined with max-pooling from the same scale then sum-pooling across different scales.
- VeryDeep [24]. [24] densely extracts multiple CNN features across multi-scales of the image with very-deep networks (16-layer and 19-layer). The features from the same scale are concatenated by sum-pooling and features from different scale are aggregated by stacking or sum-pooling. [24] also augments the test set by horizontal flipping of the images.

- Hand-crafted Features. [6] presented an Ambiguity guided Mixture Model (AMM) to integrate external context features and object features, and then used the contextualized SVM to iteratively boost the performance of object classification and detection tasks. [9] proposed an Ambiguity Guided Subcategory (AGS) mining approach to improve both detection and classification performance.

5.2. Our Setup and Parameters

It is difficult to make a completely fair comparison among different CNN based methods as the CNN configurations, the data augmentation and the pre-training could substantially influence the results. All CNN based methods can benefit from extra training data and more powerful networks as shown in [18, 26, 19, 5, 24]. To fairly evaluate our proposed framework, we develop our system based on the common 8-layer CNN pretrained on ILSVRC 2012 dataset with 1000 categories. The details of the fine tuning process has been elaborated in Section 4.3. Once the LMNN CNN and the standard CNN (see Fig. 2) are trained, the system is applied to map each training image into a final FV feature. Finally, TopPush [16] is chose to learn linear one-vs-all classifiers for each category, which produces the scores for each binary sub-problem of the multi-label datasets. The scores are then evaluated with standard VOC evaluation package. All the experiments are run on a computer with Intel i7-3930K CPU, 32G main memory and an nVIDIA Tesla K40 card.

For the proposal extraction, we employ selective search [25], which typically generates around 1500 proposals on average from every image in the PASCAL VOC 2007 dataset using the parameters suggested in [25]. Considering the computational time and the hardware limitation, we random sample around 400 proposals per image for training and testing.

For the parameters of Fisher vector, we follow [20] to first employ PCA to reduce the dimension of the original features to preserve around 90% energy. For VOC 2007 and 2012 datasets, after PCA, the standard CNN features is reduced to around 450-d. After PCA, we generate 128 GMM codewords and encode each image with IFV similar to [20].

For fine-tuning the LMNN CNN, we set the trade-off parameter $\alpha = 1$ (see (6)) and the nearest neighbor number $\hat{k} = 10$ (for training). For combining the feature view and the label view features, we select the trade-off parameter λ (specified at the end of Section 4.2) from $\{1, 0.5, 0.25\}$ by cross-validation. For the nearest neighbor number k used in testing, generally bigger k leads to better accuracy, but we observe there is no performance gain for $k > 50$. Thus, we set $k = 50$ for testing. For faster NN search, we employ FLANN [17] with "autotuned" parameters.

5.3. Image Classification Results

Image Classification on VOC 2007: Table 2 reports our experimental results compared with state-of-the-art methods on VOC 2007. In the upper part of the table we compare with the hand-crafted feature based methods and the CNN based methods pre-trained on ILSVRC 2012 using 8-layer network. To demonstrate the effectiveness of individual components, we consider three variations of our proposed framework: 'FeV', 'FeV+LV-10' and 'FeV+LV-20', where 'FeV' uses only the feature view (i.e. without the label view features), 'FeV+LV-10' uses both the feature view and the label view with 10 categories of ground-truth bounding boxes of the training set, and 'FeV+LV-20' is the one with 20 categories of ground-truth bounding boxes.

From the upper part of Table 2, we can see that using just feature view ('FeV'), we already outperform the state-of-the-art proposal-based method ('HCP-1000C') by 2.2%, which suggests that Fisher vector as a holistic representation for bags is superior than max-pooling. With all 20 categories of ground-truth bounding boxes of the training set, our multi-view framework ('FeV+LV-20') achieves a further 2.5% performance gain. This significant performance gain validates the effectiveness of the label view. Our framework shows good performance especially for difficult categories such as BOTTLE, COW, TABLE, MOTOR and PLANT.

If we just use the ground-truth bounding boxes from the first 10 categories (PLANE to COW), our framework ('FeV+LV-10') still outperforms single feature view ('FeV') by a margin of 1.3%. As expected, using the bounding boxes of the categories from PLANE to COW can boost the performance of these categories as shown in the table. However, it is interesting to see that the label view also improves the accuracies of unseen categories such as HORSE, PERSON and TV. This is mainly because the proposed label view encoding is a form of local similarity representation, which can generalize quite well to unseen categories.

In the lower part of Table 2, we list the results of 'HCP-2000C' [26], which uses additional 1000 categories from ImageNet that are semantically close to VOC 2007 categories for CNN pre-training, and 'VeryDeep' [24], which densely extracts multiple CNN features from 5 scales and combines two very-deep CNN models (16-layer an 19-layer). Our framework ('FeV+LV-20') can still outperform 'HCP-2000C', but is inferior to 'VeryDeep' since our framework is based on the common 8-layer CNN.

To demonstrate the potential of our framework, we replace the 8-layer CNNs in our framework by the 16-layer CNN model in [24], which is denoted as 'FeV+LV-20-VD'. Unlike [24], we do not use any data augmentation or multi-scale dense sampling in the feature extraction stage. Our 'FeV+LV-20-VD' outperforms 'VeryDeep' by nearly 1%. By further averaging the scores of 'VeryDeep' [24] and

Table 2. Comparisons of the classification results (in %) of state-of-the-art approaches on VOC 2007 (TRAINVAL/TEST). The upper part shows the results of the hand-crafted feature based methods and the CNN based methods trained with 8-layer CNN and ILSVRC 2012 dataset. The lower part shows the results of the methods trained with very-deep CNN or with additional training data.

	PLANE	BIKE	BIRD	BOAT	BOTTLE	BUS	CAR	CAT	CHAIR	COW	TABLE	DOG	HORSE	MOTOR	PERSON	PLANT	SHEEP	SOFA	TRAIN	TV	MAP
AGS [9]	82.2	83.0	58.4	76.1	56.4	77.5	88.8	69.1	62.2	61.8	64.2	51.3	85.4	80.2	91.1	48.1	61.7	67.7	86.3	70.9	71.1
AMM [6]	84.5	81.5	65.0	71.4	52.2	76.2	87.2	68.5	63.8	55.8	65.8	55.6	84.8	77.0	91.1	55.2	60.0	69.7	83.6	77.0	71.3
CNN-SVM [23]	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.3	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.4	71.8	73.9
PRE-1000C [18]	88.5	81.5	87.9	82.0	47.5	75.5	90.1	87.2	61.6	75.7	67.3	85.5	83.5	80.0	95.6	60.8	76.8	58.0	90.4	77.9	77.7
HCP-1000C [26]	95.1	90.1	92.8	89.9	51.5	80.0	91.7	91.6	57.7	77.8	70.9	89.3	89.3	85.2	93.0	64.0	**85.7**	62.7	94.4	78.3	81.5
FeV	93.3	92.8	91.8	86.5	57.5	84.3	93.7	90.6	64.7	78.9	74.1	90.3	91.1	90.7	95.7	67.4	82.4	70.8	94.3	83.3	83.7
FeV+LV-10	**96.6**	93.6	**94.0**	**89.5**	59.6	87.4	**94.8**	91.0	67.3	81.4	76.3	91.0	**93.5**	91.4	96.1	64.2	83.4	68.5	95.8	84.0	85.0
FeV+LV-20	95.7	**94.6**	93.9	87.8	**62.7**	**87.9**	**94.8**	**92.2**	**67.5**	**82.4**	**77.8**	**92.0**	93.2	**92.2**	**97.1**	**72.5**	85.3	**73.4**	**96.2**	**85.5**	**86.2**
HCP-2000C [26]	96.0	92.1	93.7	93.4	58.7	84.0	93.4	92.0	62.8	89.1	76.3	91.4	95.0	87.8	93.1	69.9	90.3	68.0	96.8	80.6	85.2
VeryDeep [24]	**98.9**	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	87.8	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
FeV+LV-20-VD	97.9	**97.0**	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	**97.7**	**95.9**	**98.6**	77.6	88.7	78.0	98.3	89.0	90.6
Fusion	98.2	96.9	**97.1**	**95.8**	**74.3**	**94.2**	**96.7**	**96.7**	**76.7**	**90.5**	**88.0**	**96.9**	**97.7**	**95.9**	**98.6**	**78.5**	**93.6**	**82.4**	**98.4**	**90.4**	**92.0**

Table 3. Comparisons of the classification results (in %) of state-of-the-art approaches on VOC 2012 (TRAINVAL/TEST). The upper part shows the results of the hand-crafted feature based methods and the CNN based methods trained with 8-layer CNN and ILSVRC 2012 dataset. The lower part shows the results of the methods trained with very-deep CNN or with additional training data.

	PLANE	BIKE	BIRD	BOAT	BOTTLE	BUS	CAR	CAT	CHAIR	COW	TABLE	DOG	HORSE	MOTOR	PERSON	PLANT	SHEEP	SOFA	TRAIN	TV	MAP
NUS-PSL [26]	97.3	84.2	80.8	85.3	60.8	89.9	**86.8**	89.3	**75.4**	77.8	75.1	83.0	87.5	90.1	95.0	57.8	79.2	**73.4**	94.5	80.7	82.2
PRE-1000C [18]	93.5	78.4	87.7	80.9	57.3	85.0	81.6	89.4	66.9	73.8	62.0	89.5	83.2	87.6	95.8	61.4	79.0	54.3	88.0	78.3	78.7
HCP-1000C [26]	**97.7**	83.0	**93.2**	87.2	59.6	88.2	81.9	94.7	66.9	81.6	68.0	**93.0**	88.2	87.7	92.7	59.0	85.1	55.4	93.0	77.2	81.7
FeV	96.8	87.8	88.7	87.2	63.8	**92.3**	86.2	92.3	72.4	82.0	76.0	91.9	90.3	90.3	95.2	61.2	82.6	65.6	92.8	84.4	84.0
FeV-LV-10	97.3	**89.1**	**91.5**	**88.5**	**66.7**	92.2	**87.2**	94.0	74.0	82.7	77.8	91.6	91.1	92.7	95.7	66.5	85.4	**69.4**	**95.6**	85.4	85.7
FeV-LV-20	**97.4**	88.9	91.2	87.4	64.2	92.2	86.4	**95.0**	**75.1**	**84.6**	**78.7**	**93.1**	**91.9**	**93.1**	**96.6**	**67.3**	**86.2**	**69.4**	95.3	**85.8**	**86.0**
PRE-1512C [18]	94.6	82.9	88.2	84.1	60.3	89.0	84.4	90.7	72.1	86.8	69.0	92.1	93.4	88.6	96.1	64.3	86.6	62.3	91.1	79.8	82.8
HCP-2000C [26]	97.5	84.3	93.0	89.4	62.5	90.2	84.6	94.8	69.7	90.2	74.1	93.4	93.7	88.8	93.3	59.7	90.3	61.8	94.4	78.0	84.2
[19]	96.7	88.8	92.0	87.4	64.7	91.1	87.4	94.4	74.9	89.2	76.3	93.7	95.2	91.1	97.6	66.2	91.2	70.0	94.5	83.7	86.3
VeryDeep [24]	99.0	89.1	**96.0**	**94.1**	74.1	92.2	85.3	**97.9**	79.9	92.0	**83.7**	**97.5**	96.5	94.7	97.1	63.7	**93.6**	75.2	97.4	87.8	89.3
NUS-HCP-AGS [26]	**99.0**	91.8	94.8	92.4	72.6	**95.0**	**91.8**	97.4	**85.2**	**92.9**	83.1	96.0	96.6	**96.1**	94.9	68.4	92.0	**79.6**	97.3	88.5	90.3
FeV+LV-20-VD	98.4	92.8	93.4	90.7	74.9	93.2	90.2	96.1	78.2	89.8	80.6	95.7	96.1	95.3	97.5	73.1	91.2	75.4	97.0	88.2	89.4
Fusion	98.9	**93.1**	**96.0**	**94.1**	**76.4**	93.5	90.8	**97.9**	80.2	92.1	82.4	97.2	**96.8**	95.7	**98.1**	**73.9**	**93.6**	76.8	**97.5**	**89.0**	**90.7**

'FeV+LV-20-VD' (denoted as 'Fusion'), we achieve state-of-the-art mAP of 92.0%. This suggests that our proposal-based framework and the multi-scale CNN extracted from the whole image are complement to each other.

Image Classification on VOC 2012: Table 3 reports our experimental results compared with those of the state-of-the-art methods on VOC 2012. Similar to Table 2, we compare with the hand-crafted feature based methods and the CNN based methods pre-trained on ILSVRC 2012 using 8-layer CNN model in the upper part and the methods trained with additional data or very-deep CNN models in the lower part.

The results are consistent with those on VOC 2007. Our framework that uses only the feature view ('FeV') already outperforms the state-of-the-art hand-crafted feature method ('NUS-PSL') by 1.8% and the state-of-the-art proposal-based CNN method (HCP-1000C) by 2.3%. With the aid of the label view, our 'FeV+LV-20' obtains an additional 2% performance again, even outperforming the two proposal-based methods pre-trained on additional 512 or 1000 categories of image data ('PRE-1512C' and 'HCP-2000C') and comparable to [19]. By employing just 10 categories of bounding boxes, the mAP performance of our 'FeV+LV-10' does not degrade much.

When employed with the very-deep 16-layer CNN model [24], our framework ('FeV+LV-20-VD') achieves similar performance as 'VeryDeep'. When we averagely fuse the scores of [24] and our proposal-based representation, our method ('Fusion') achieves state-of-the-art result, outperforming [26].

6. Conclusion

In this paper, we have proposed a multi-view multi-instance framework for solving the multi-label classification problem. Compared with existing works, our framework makes use of the strong labels to provide another view of local information (label view) and combines it with the typical feature view information to boost the discriminative power of feature extraction for multi-label images. The experimental results validates the discriminative power and the generalization ability of the proposed framework.

For future directions, there are several possibilities to explore. First of all, we can improve the scalability and possibly also the performance of the framework by establishing a proposal selection criteria to filter out noisy proposals. Secondly, we may build a suitable candidate pool directly from the extracted proposals to eliminate the need for strong labels.

Acknowledgments This research is supported by Singapore MoE AcRF Tier-1 Grant RG138/14 and also partially supported by the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS 15*, pages 561–568. MIT Press, 2003. 3, 4

[2] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marqués, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 3

[3] A. Bellet, A. Habrard, and M. Sebban. A survey on metric learning for feature vectors and structured data. *CoRR*, abs/1306.6709, 2013. 3, 4

[4] E. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4:888–900, 1992. 4

[5] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 2, 5, 6, 7

[6] Q. Chen, Z. Song, J. Dong, Z. Huang, Y. Hua, and S. Yan. Contextualizing object detection and classification. *IEEE TPMAI*, 37(1):13–27, 2015. 7, 8

[7] Y. Chen, J. Bi, and J. Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1931–1947, Dec. 2006. 3

[8] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014. 3, 6

[9] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, and S. Yan. Subcategory-aware object classification. In *CVPR*, pages 827–834. IEEE, 2013. 7, 8

[10] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010. 6

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, Sept. 2010. 4

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2, 3

[13] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(6):607–616, June 1996. 3

[14] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, September 2014. 3

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS 25*, pages 1097–1105. Curran Associates, Inc., 2012. 1, 4, 6

[16] N. Li, R. Jin, and Z.-H. Zhou. Top rank optimization in linear time. In *NIPS 27*, pages 1502–1510. 2014. 7

[17] M. Muja and D. G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE PAMI*, 36, 2014. 7

[18] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 1, 2, 6, 7, 8

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Weakly supervised object recognition with convolutional neural networks. Technical Report HAL-01015140, INRIA, 2014. 1, 2, 3, 6, 7, 8

[20] F. Perronnin, J. Snchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 4, 7

[21] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013. 4

[22] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. 1, 2, 4, 6

[23] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, June 2014. 1, 2, 4, 6, 8

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3, 6, 7, 8

[25] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *ICCV*, 2013. 2, 3, 7

[26] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. CNN: single-label to multi-label. *CoRR*, abs/1406.5726, 2014. 1, 2, 3, 6, 7, 8

[27] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, June 2009. 3, 4, 5

[28] J. W. Xiu-Shen Wei and Z. hua Zhou. Scalable multi-instance learning. In *ICDM*, 2014. 4

[29] X. Xu and E. Frank. Logistic regression and boosting for labeled bags of instances. In *PAKDD*, pages 272–281. Springer-Verlag, 2004. 3

[30] L. Yang, R. Jin, R. Sukthankar, and Y. Liu. An efficient algorithm for local distance metric learning. In *AAAI*, pages 543–548, 2006. 3

[31] A. Yu and K. Grauman. Predicting useful neighborhoods for lazy local learning. In *NIPS 27*, pages 1916–1924, 2014. 2

[32] Q. Zhang, G. Hua, W. Liu, Z. Liu, and Z. Zhang. Can visual recognition benefit from auxiliary information in training?. In *ACCV*, 2014. 3

[33] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *ICML*, pages 1249–1256, 2009. 3

[34] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 3