Multi-Object Classification and Unsupervised Scene Understanding Using Deep Learning Features and Latent Tree Probabilistic Models

Anonymous CVPR submission

Paper ID ****

1. Introduction

Deep learning has shown state-of-art classification performance on datasets such as ImageNet, which contain a single object in each image. However, multi-object classification is far more challenging. Independent binary classifiers ignore the relationships between labels and miss the opportunity to transfer and share knowledge among different label categories during learning. In this work, we propose an efficient multi-object classification framework by incorporating contextual information in images. The context in natural images captures relationships between various object categories, such as co-occurrence of objects within a scene or relative positions of objects with respect to a background scene. Incorporating such contextual information can vastly improve detection performance, eliminate false positives, and provide a coherent scene interpretation.

We incorporate contextual information in natural images through a conditional latent tree probabilistic model (CLTM), where the object co-occurrences are conditioned on the transferable features \([8, 6]\) (fc7 features) from pre-trained Imagenet \([4]\) CNN as input. We learn the CLTM tree structure using conditional pairwise probabilities for object co-occurrences, estimated through kernel methods, and we learn its node and edge potentials by training a new 3-layer neural network, which takes fc7 features as input. Algo.1 gives overview of our framework. Object classification is carried out via inference on the learnt conditional tree model using message passing algorithm.

2. Proposed Framework

Our model predicts a structured output \(y \in \{0, 1\}^L\) for extracted fc7 feature \(x\) of an image. To achieve this goal, we use a dependency structure that relates different object labels. We model this dependency structure using a latent tree that allow for more complex structures of dependence compared to a fully observed tree.

Learning Latent Tree Structure: We use kernel conditional embedding framework, described in [7] to estimate conditional pairwise probabilities for learning the latent tree.

Algorithm 1 Overview of the Framework

Require: Labeled image-set \(\mathcal{I} = \{(I^1, y^1), \cdots, (I^n, y^n)\}\)
1: \(\{x^1, x^2, \cdots, x^n\} \leftarrow \text{ExtractFc7Features(}\mathcal{I}\text{)}\)
2: Estimate conditional distance matrix : 
   \(D \leftarrow \text{CondDistanceMatrix}(\{(x^1, y^1), \cdots, (x^n, y^n)\})\) using kernel methods.
3: Extract tree structure using [2]
   \(T \leftarrow \text{CLRG}(D)\)
4: Training a NN with randomly initialized weights \(W:\)
5: \textbf{repeat} 
6:  \hspace{1em} randomly select a mini-batch \(M\).
7:  \hspace{1em} compute negative marginalized log-likelihood loss: 
   \(\text{Eqn.}(1)\)
   \(\mathcal{L} \leftarrow \text{Loss}(W, T, M)\)
8:  \hspace{1em} \(W \leftarrow \text{BackpropogateGradient}(\mathcal{L})\)
9: \textbf{until} convergence
10: Given a test image \(T: x^t \leftarrow \text{ExtractFc7Features}(T)\)
11: \hspace{1em} Potentials \(\leftarrow \text{FeedForward}(W, x^t)\)
12: Prediction: \(y \leftarrow \text{arg min}_y \text{Energy}(Y, \text{Potentials})\)

structure. For our conditional setting, we use the following form of the distance function:

\[
\hat{d}_{kt} = \frac{1}{n} \sum_{i=1}^{n} \log \left( \frac{|\text{det}(\mathbb{E}[Y_k \otimes Y_i | X = x^t])|}{\sqrt{S_{k,k} \cdot S_{t,t}}} \right),
\]

where \(S_{k,k} := |\text{det}(\mathbb{E}[Y_k \otimes Y_k | X = x^t])|\), and similarly for \(S_{t,t}\), for observed nodes \(k, t\) using \(N\) samples. We employ the CL grouping [2] to learn the tree structure from the estimated distances.

Training CLTM using Neural Networks: Once we recover the structure \(T = (\mathcal{Z}, \mathcal{E})\), we use conditional latent tree model to model \(P(\mathcal{Z}|X)\). We model unnormalized probability of \(\mathcal{Z}\) as exponential of negative of energy function defined using the below Eqn.

\[
\mathcal{E}(x, z, \theta) = \sum_{k \in \mathcal{Z}} \phi_k(x, \theta)z_k + \sum_{(k, t) \in \mathcal{E}} \phi_{(k,t)}(x, \theta)z_kz_t
\]
\( \phi_k(x, \theta) \) and \( \phi_{(k,t)}(x, \theta) \) indicate the node and edge potentials respectively. Instead of restricting the potentials to linear functions of covariates, we generalize potentials as functions represented by outputs of a neural network. For a given neural network architecture, weights are learned by backpropagating the gradient through negative marginal log-likelihood loss (1).

\[
\mathcal{L} = \mathbb{E}[\mathcal{E}(W, y, x, h) | y, x] - \mathbb{E}[\mathcal{E}(W, y, x, h) | x]
\] (1)

And the gradient is evaluated using below Eqn.

\[
\frac{\partial \mathcal{L}}{\partial W} = \mathbb{E}\left[\frac{\partial \mathcal{E}(W, y, x, h)}{\partial W} | x, y\right] - \mathbb{E}\left[\frac{\partial \mathcal{E}(W, y, x, h)}{\partial W} | x\right]
\]

3. Experimental Results

We did experiments to evaluate our model by using the non- iconic image data-set MS COCO [5]. We use Caffe [3] to extract fc7 features and use them as input to a 3 layer neural network. The outputs of this neural network correspond to node potentials of the energy function of CLTM model. To avoid over-fitting we make edge potentials independent of input covariates. We use stochastic gradient descent and compute the gradient of loss function defined in Eqn.(1) to train the model. We use an independent classifier trained using 3 layer neural network (Indep. Classifier) as a baseline, and compare precision-recall measures with our proposed CLTM. Note that most training images contain fewer than 3 instances of different object categories.

**Classification Performance**: For 3 layer neural network independent classifier, we use a threshold of 0.5 to make binary decisions for different object labels. For CLTM, we use the MAP configuration to make binary decisions. The overall relative gain in F-measure for our method is 7%. For difficult objects like couch, frisbee, cup, bowl, remote, fork, and wine-glass, the F-measure relative gain is 41%, 48%, 50%, 53%, 113%, 122%, and 171% respectively. We see across the board improvement for all object categories over the entire precision-recall curve.

**Role of Latent Nodes in Scene Classification**: We observe that latent nodes capture high-level semantic information common to images, based on the neighborhoods of object categories in the latent tree. When we consider the top images with largest activations of node potential for a given latent node, we find diverse images with different objects, but with a unifying common theme. For instance, for one of the latent variables, the top images capture a grass-land scene but with different animals in different images. Similarly, the latent variable representing an outdoor scene contains diverse images with traffic, beaches, and buildings. We also quantitatively show that the latent variables yield efficient scene classification performance on the MIT-Indoor dataset, without any re-training, and without using any scene labels during training. We use the marginal probabilities of the latent variables in our model on test images, and perform \( k \)-means clustering. For validation, we match these clusters to ground truth scene categories using maximum weight matching [1]. We obtain 20% improvement in misclassification rate of the scenes, compared to the \( k \)-means clustering on outputs of baseline model.

4. Discussion

We propose a unified framework for multi-object classification and scene understanding that combines the strengths of multiple machine learning techniques, deep learning, probabilistic models, and kernel methods. We demonstrate significant improvement over state-of-art deep learning methods, especially on challenging objects. We learn a conditional latent tree model, where we condition on pre-trained deep learning features. We employ kernel methods to learn the structure of the hierarchical tree model, and we train a new smaller neural network to learn the node and edge potentials of the model. Multi-object classification is carried out via inference on the tree. All these steps are efficient and scalable to large datasets with a large number of object categories.

References