Top-down Learning for Structured Labeling with Convolutional Pseudoprior

Saining Xie[1†], Xun Huang[2†], Zhuowen Tu[1]

[1]Dept. of CogSci and Dept. of CSE, UC San Diego
{s9xie,ztu}@ucsd.edu
[2]Dept. of Computer Science, Cornell University
xh258@cornell.edu

Abstract. Current practice in convolutional neural networks (CNN) remains largely bottom-up and the role of top-down process in CNN for pattern analysis and visual inference is not very clear. In this paper, we propose a new method for structured labeling by developing convolutional pseudoprior (ConvPP) on the ground-truth labels. Our method has several interesting properties: (1) compared with classic machine learning algorithms like CRFs and Structural SVM, ConvPP automatically learns rich convolutional kernels to capture both short- and long-range contexts; (2) compared with cascade classifiers like Auto-Context, ConvPP avoids the iterative steps of learning a series of discriminative classifiers and automatically learns contextual configurations; (3) compared with recent efforts combining CNN models with CRFs and RNNs, ConvPP learns convolution in the labeling space with improved modeling capability and less manual specification; (4) compared with Bayesian models like MRFs, ConvPP capitalizes on the rich representation power of convolution by automatically learning priors built on convolutional filters. We accomplish our task using pseudo-likelihood approximation to the prior under a novel fixed-point network structure that facilitates an end-to-end learning process. We show state-of-the-art results on sequential labeling and image labeling benchmarks.

Keywords: Structured Prediction, Deep Learning, Semantic Segmentation, Top-down Processing

1 Introduction

Structured labeling is a key machine learning problem: structured inputs and outputs are common in a wide range of machine learning and computer vision applications [1,2,3]. The goal of structured labeling is to simultaneously assign labels (from some fixed label set) to individual elements in a structured input. Markov random fields (MRFs) [4] and conditional random fields (CRFs) [2] have been widely used to model the correlations between the structured labels. However, due to the heavy computational burden in their training and

---

† equal contribution.

testing/inference stages, MRFs and CRFs are often limited to capturing a few neighborhood interactions with consequent restrictions of their modeling capabilities. Structural SVM methods [5] and maximum margin Markov networks (M^3N) [6] capture correlations in a way similar to CRFs, but they try to specifically maximize the prediction margin; these approaches are likewise limited in the range of contexts, again due to associated high computational costs. When long range contexts are used, approximations are typically used to trade between accuracy and efficiency [7]. Other approaches to capture output variable dependencies have been proposed by introducing classifier cascades. For example, cascade models [8,9,10] in the spirit of stacking [11], are proposed to take the outputs of classifiers of the current layer as additional features for the next classifiers in the cascade. Since these approaches perform direct label prediction (in the form of functions) instead of inference as in MRFs or CRFs, the cascade models [8,9] are able to model complex and long-range contexts.

Despite the efforts in algorithmic development with very encouraging results produced in the past, the problem of structured labeling remains a challenge. To capture high-order configurations of the interacting labels, top-down information, or prior offers assistance in both training and testing/inference. The demonstrated role of top-down information in human perception [12,13,14] provides a suggestive indication of the form that top-down information could play in structured visual inference. Systems trying to explicitly incorporate top-down information under the Bayesian formulation point to a promising direction [15,16,17,18] but in the absence of a clear solution. Conditional random fields family models that learn the posterior directly [2,8,9,19] alleviates some burdens on learning the labeling configuration, but still with many limitations and constraints. The main difficulty is contributed by the level of complexity in building high-order statistics to capture a large number of interacting components within both short- and long- range contexts.

From a different angle, building convolutional neural networks for structured labeling [20] has resulted in systems that greatly outperform many previous algorithms. Recent efforts in combining CNN with CRF and RNN models [21,22] have also shed light onto the solution of extending CNN to structured prediction. However, these approaches still rely on CRF-like graph structure with limited neighborhood connections and heavy manual specification. More importantly, the explosive development in modeling data using layers of convolution has not been successfully echoed in modeling the prior in the label space.

In this paper, we propose a new structured labeling method by developing convolutional pseudoprior (ConvPP) on the ground-truth labels, which is infeasible by directly learning convolutional kernels using the existing CNN structure. We accomplish our task by developing a novel end-to-end fixed-point network structure using pseudo-likelihood approximation [23] to the prior that learns convolutional kernels and captures both the short- and the long- range contextual labeling information. We show state-of-the-art results on benchmark datasets in sequential labeling and popular image labeling.

2   Related Work

We first summarize the properties of our proposed convolutional pseudoprior (ConvPP) method: (1) compared with classical machine learning algorithms like CRFs [2], Structural SVM ([5]), and max-margin Markov networks [6] , ConvPP automatically learns rich convolutional kernels to capture both the short- and the long- range contexts. (2) Compared with cascade classifiers [8,9], ConvPP avoids the time-consuming steps of iteratively learning a series of discriminative classifiers and it automatically learns the contextual configurations (we have tried to train a naive auto-context type of fully convolutional model instead of modeling prior directly from the ground-truth label space but without much success; the overall test error did not decrease after long-time training with many attempts of parameter tweaking; this is possibly due to the difficulty in capturing meaningful contexts on the predicted labels, which are noisy). (3) Compared with recent efforts combining CNN models with CRFs and RNNs [21,22], ConvPP learns convolution in the labeling space with improved modeling capability and less manual specification. (4) Compared with Bayesian models [16,24] ConvPP capitalizes on the rich representation power of CNN by automatically learning convolutional filters on the prior.

In addition, we will discuss some other related work. [25] addresses structured (image) labeling tasks by building a multi-scale CRF with handcrafted features and constrained context range, whereas in our work we learn the context automatically in an end-to-end network. [26] also combines RBM and CRF to incorporate shape prior for face segmentation. [19] is able to learn a large neighborhood graph but under a simplified model assumption; in [27] deep convolutional networks are learned on a graph but the focus there is not for structured labeling; deep belief nets (DBN) [28] and auto-encoders [29] are generative models that potentially can be adapted for learning the prior but it a clear path for structured labeling is lacking. Our work is also related to recurrent neural networks (RNNs) [1], but ConvPP has its particular advantage in: (1) modeling capability as explicit convolutional features are learned on the label space; (2) reduced training complexity as the time-consuming steps of computing recurrent responses are avoided by directly using the ground truth labels as a fixed-point model. In deep generative stochastic networks[30], pseudo-likelihood is used to train a deep generative model, but not for learning priors with CNN.

To summarize, ConvPP builds an end-to-end system by learning a novel hybrid model with convolutional pseudopriors on the labeling space and traditional bottom-up convolutional neural networks for the appearance.

3   Formulations

We first briefly discuss the structured labeling problem and understand it from the Bayesian point of view. Let $\mathcal{X}$ be the space of input observations and $\mathcal{Y}$ be the space of possible labels. Assume any data-label pairs $(\mathbf{X}, \mathbf{Y})$ follow a joint distribution $p(\mathbf{X}, \mathbf{Y})$. We seek to learn a mapping $F : \mathcal{X} \to \mathcal{Y}$ that minimizes

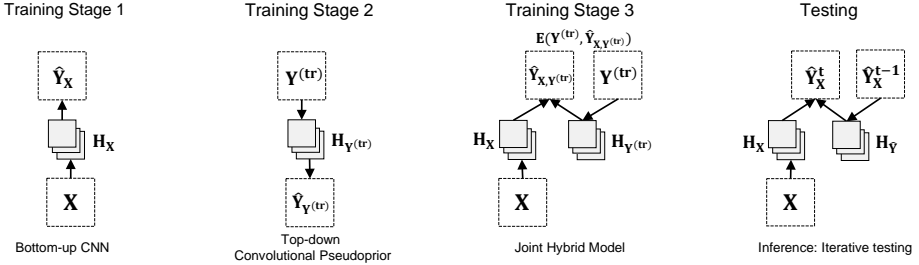| Training Stage 1	Training Stage 2	Training Stage 3	Testing

Fig. 1: The architecture of our ConvPP framework. At the first training stage, we train a bottom-up CNN model with image data as input; at the second training stage, we train a top-down convolutional pseudoprior model from ground-truth label maps. The hidden representations are then concatenated and the network is fine-tuned with the joint hybrid model. At inference, since we don't have the ground-truth label anymore, we iteratively feed predictions to the convolutional pseudoprior part.

the expected loss. For a new input sample $X \in \mathcal{X}$, we want to determine the optimal labeling $\mathbf{Y}^*$ that maximizes the posterior probability $p(\mathbf{Y}|\mathbf{X})$.

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} p(\mathbf{Y}|\mathbf{X}) = \arg\max_{\mathbf{Y}} p(\mathbf{X}|\mathbf{Y})p(\mathbf{Y}) \qquad (1)$$

In the scenario of structured labeling such as pixel-wise labeling, intuitively, the labeling decision should be made optimally by considering both the appearance and the prior, based on the Bayes rule in Eq. (1). However, learning both $p(\mathbf{X}|\mathbf{Y})$ and $p(\mathbf{Y})$ for complex structures is considered as very challenging. Our motivation here is to capitalize on the rich representational and compelling computational power of CNN for modeling both the appearance and prior. A large amount of work in the past using CNN has been primarily focused on training strong classifiers for predicting semantic labels (a discriminative way of modeling the appearance, [20]), but rarely on the prior part (top-down information).

To formulate our structured labeling problem, here we consider a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In a 1-D sequential labeling case, the graph is equivalent to a chain. The edge set $\mathcal{E}$ decides the graph topology, and hence the neighborhoods of every node. We denote $\mathcal{N}_i/i$ as the neighborhoods of node v_i. For each node v_i, we have its associated data $\mathbf{x}_i$, ground-truth label y_i, and ground-truth labels for all the neighborhoods of v_i as $\mathbf{y}_{\mathcal{N}_i/i}$. Inspired by pseudo-likelihood [23] and the hybrid model in [31], we make an approximation to the posterior in Eq. (1) as follows:

$$p(\mathbf{Y}|\mathbf{X}) \propto p(\mathbf{Y})p(\mathbf{X}|\mathbf{Y}) \dot{\propto} \prod_i p(y_i|\mathbf{Y}_{\mathcal{N}_i/i}) \cdot \prod_i p(y_i|\mathbf{X}) \qquad (2)$$

where $\mathbf{Y}_{\mathcal{N}_i/i}$ encodes a neighborhood structure (contexts) of y_i for computing a pseudo-likelihood $p(y_i|\mathbf{Y}_{\mathcal{N}_i/i})$ [23] to approximate $p(\mathbf{Y})$, *but now as a prior*.

In addition, to see how the approximation to $p(\mathbf{X}|\mathbf{Y})$ by $\prod_i p(y_i|\mathbf{X})$ is obtained from Bayesian to conditional probability: (1) assume pseudo-likelihood

on each pixel i to approximate the likelihood term $p(\mathbf{X}|\mathbf{Y})$, using $p(\mathbf{x}_i|x_{\mathcal{N}_i/i}, Y)$. Note that $\mathbf{x}_{\mathcal{N}_i/i}$ includes all the neighboring pixels of pixel i but excluding i; (2) assume independence, to approximate $p(\mathbf{x}_i|\mathbf{x}_{\mathcal{N}_i/i}, Y)$ by $p(\mathbf{x}_i|\mathbf{x}_{\mathcal{N}_i/i}, y_i)$; (3) $p(\mathbf{x}_i|\mathbf{x}_{\mathcal{N}_i/i}, y_i) = p(\mathbf{x}_i, y_i|\mathbf{x}_{\mathcal{N}_i/i})/P(y_i|\mathbf{x}_{\mathcal{N}_i/i}))$ and drop $p(y_i|\mathbf{x}_{\mathcal{N}_i/i})$ for another approximation. This leads to $p(\mathbf{x}_i, y_i|\mathbf{x}_{\mathcal{N}_i/i})$ which is $p(y_i|\mathbf{x}_{\mathcal{N}_i/i}, \mathbf{x}_i)p(\mathbf{x}_i|\mathbf{x}_{\mathcal{N}_i/i})$; (4) the above becomes $p(y_i|\mathbf{x}_{\mathcal{N}_i}) = p(y_i|\mathbf{X})$ when dropping $p(\mathbf{x}_i|\mathbf{x}_{\mathcal{N}_i/i})$.

This hybrid model is of special interest to us since: (1) our end-to-end deep learning framework allows a discriminative convolutional neural network (CNN) to be trained to compute $p(y_i|\mathbf{X})$ to model the appearance; (2) by directly working on the ground-truth labels $\mathbf{Y}$, we also learn a convolutional pseudoprior as $\prod_i p(y_i|\mathbf{Y}_{\mathcal{N}_i/i})$ using a pseudo-likelihood approximation.

Given a training data pair $p(\mathbf{X}, \mathbf{Y}^{(tr)})$, to solve an approximated MAP problem with convolutional pseudoprior,

$$\mathbf{Y}^{(tr)} = \arg\max_{\mathbf{Y}} \prod_i p(y_i|\mathbf{Y}_{\mathcal{N}_i/i}; \mathbf{w}_2) \cdot \prod_i p(y_i|\mathbf{X}; \mathbf{w}_1) \tag{3}$$

From another perspective, the above learning/inference scheme can be motivated by the fixed-point model [32]. Denote $\mathbf{Q}$ as the one-hot encoding of labeling $\mathbf{Y}$, and therefore $\mathbf{Q}^{(tr)}$ as the one-hot encoding of ground-truth training labeling $\mathbf{Y}^{(tr)}$. The fixed-point model solve the problem with the formulation for a prediction function $\mathbf{f}$,

$$\mathbf{Q} = \mathbf{f}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n, \mathbf{Q}; \mathbf{w}) \tag{4}$$

where $\mathbf{f}(\cdot) = [f(\mathbf{x}_1, \mathbf{Q}_{\mathcal{N}_1}; \mathbf{w}), \cdots, f(\mathbf{x}_n, \mathbf{Q}_{\mathcal{N}_n}; \mathbf{w})]^T$, $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n]^T$, and $\mathbf{q}_i = f(\mathbf{x}_i, \mathbf{Q}_{\mathcal{N}_i})$. $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2)$. To get the labeling of a structured input graph $\mathcal{G}$, one can solve the non-linear system of equations $\mathbf{Q} = \mathbf{f}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n, \mathbf{Q}; \mathbf{w})$, which is generally a very difficult task. However, [32] shows that in many cases we can assume $\mathbf{f}$ represents so called contraction mappings, and so have an attractive fixed-point (a "stable state") for each structured input. When using the ground-truth labeling in the training process, that ground-truth labeling $\mathbf{Q}^{(tr)}$ is assumed to be the stable state: $\mathbf{Q}^{(tr)} = \mathbf{f}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n, \mathbf{Q}^{(tr)}; \mathbf{w})$.

Next, we discuss the specific network architecture design and our training procedure. The pipeline of our framework is shown in Figure 1 consisting of three stages: (1) training $\mathbf{w}_1$ for $p(y_i|\mathbf{X}; \mathbf{w}_1)$; (2) training $\mathbf{w}_2$ for $p(y_i|\mathbf{Y}_{\mathcal{N}_i/i}; \mathbf{w}_2)$; and (3) fine-tuning for $\prod_i p(y_i|\mathbf{Y}_{\mathcal{N}_i/i}; \mathbf{w}_2) \cdot p(y_i|\mathbf{X}; \mathbf{w}_1)$ jointly.

At the first stage, we independently train a standard bottom-up CNN on the input data, in our work, we are especially interested in end-to-end architectures such as FCN [20]. Without loss of generality, we abstractly let the feature representations learned by FCN be $\mathbf{H_X}$, and network predictions be $\hat{\mathbf{Y}}_{\mathbf{X}}$, The error is computed with respect to the ground-truth label $\mathbf{Y}^{(tr)}$ and back-propagated during training. Similarly, at the second stage, we train a convolutional pseudoprior network on the ground-truth label space. Conceptually the prior modeling is a top-down process. Implementation-wise, the ConvPP network is still a CNN. However, the most notable difference compared with a traditional CNN

is that, the ground-truth labels are not only used as the supervision for back-propagation, but also used as the network input. We learn hidden representation $\mathbf{H}_{\mathbf{Y}^{(\mathbf{tr})}}$ and aim to combine this with the hierarchical representation $\mathbf{H}_{\mathbf{X}}$ learned in the bottom-up CNN model. Thus, combining pre-trained bottom-up CNN network and top-down ConvPP network, we build a joint hybrid model network in the third training stage. We concatenate $\mathbf{H}_{\mathbf{X}}$ and $\mathbf{H}_{\mathbf{Y}}$ (which can be fine-tuned) and learn a new classifier on top to produce the prediction $\hat{\mathbf{Y}}_{\mathbf{X},\mathbf{Y}^{(\mathbf{tr})}}$. The joint network is still trained with back-propagation in an end-to-end fashion.

At inference time, since we do not have the ground-truth label $\mathbf{Y}^{(tr)}$ available anymore, we follow the fixed-point motivation discussed above. We iteratively feed predictions $\hat{\mathbf{Y}}_{\mathbf{X}}^{\mathbf{t-1}}$ made at previous iteration, to the ConvPP part of the hybrid model network. The starting point $\hat{\mathbf{Y}}_{\mathbf{X}}^{\mathbf{0}} = \mathbf{0}$ can be a zero-initialized dummy prediction, or we can simply use $\hat{\mathbf{Y}}_{\mathbf{X}}^{\mathbf{0}} = \hat{\mathbf{Y}}_{\mathbf{X}}$ given the pre-trained bottom-up CNN model.

This conceptually simple approach to approximate and model the prior naturally faces two challenges: 1) How do we avoid trivial solutions and make sure the ConvPP network can learn meaningful structures instead of converging to an identity function? 2) When the bottom-up CNN is deep and involves multiple pooling layers, how to match the spatial configurations and make sure that, $\mathbf{H}_{\mathbf{X}}$ and $\mathbf{H}_{\mathbf{Y}^{(\mathbf{tr})}}$ are compatible in terms of the appearances and structures they learn.

ConvPP network architectures. We will now explain the architecture design in ConvPP network to address the issues above. We have explored possible ways to avoid learning a trivial solution. Besides the ConvPP architecture design, one might think learning a convolutional auto-encoder on the ground-truth label space can achieve similar goal. However, we found that when training an auto-encoder on label space, the problem of trivial recovery is even more severe compared to training auto-encoders on natural images. We tried different regularization and sparsification techniques presented in recent convolutional auto-encoder works (e.g. [33]), but none of them work in our case. See Figure 2 for a visual comparison. We conjecture that the reasons could be (1) the ground-truth labels are much simpler in their appearances, compared with natural images with rich details. Thus the burden of being identically reconstructed is greatly eased; (2) on the other hand, the structures like class inter-dependencies, shape context and relative spatial configurations are highly complex and subtle, make it really challenging to learn useful representations.

Donut filter. Here we use a very simple yet effective approach to conquer the issues: our ConvPP network contains only a single convolutional layer, where we apply filters referred as "donut filters". The name comes from the way we modify the traditional convolution kernels: we make a hole in the center of the kernel. Figure 2 shows an example where a 3 by 3 hole is in the middle of a 7 by 7 convolution filter. Given that we only have one convolutional layer, we impose a hard constraint on the ConvPP representation learning process: the reconstruction of the central pixel label will never see its original value, instead
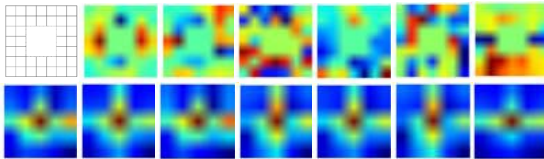
Fig. 2: The first row shows the donut filter we use for training the ConvPP network, and examples of learned real-valued filters after the ConvPP network training, originally initialized randomly. Note that a square hole (showing in light-green color) in the center is kept zero-valued during training, and it enforces the filter to learn non-trivial transformations. The second row shows trivial filters (close to identity transformation) are learned by a conventional auto-encoder.

it can only be inferred from the neighboring labels. This is aligned with our pseudoprior formulation in Eq. (3).

Donut filters are not supposed to be stacked to form a deep variant, since the central pixel label information, even though cropped from one layer, can be propagated from lower layers which enables the network to learn a trivial solution. Empirically we found that one hidden convolution layer with multiple filters is sufficient to approximate and model the useful prior in the label space.

Multi-scale ConvPP. A natural question then becomes, since there is only one convolution layer in the ConvPP network, the receptive field size is effectively the donut filter kernel size. Small kernel size will result in very limited range of context that can be captured, while large kernel size will make the learning process extremely hard and often lead to very poor local minimum.
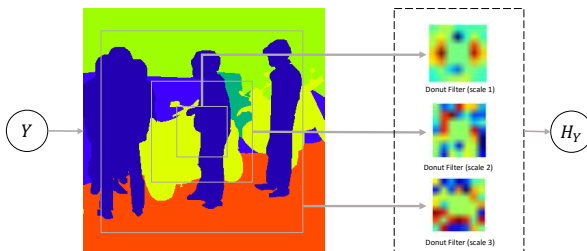

Fig. 3: Multiple donut filter layers with the same kernel size are integrated into different depth of the joint network. Multi-scale context learning is naturally handled.

To combat this issue, as illustrated in Figure 3, we leverage the unique simplicity of ground-truth labeling maps, and directly downsample the input ground-truth maps through multiple pooling layers to form a chain of multi-scale ground-truth maps. The in-network pooling layers also keep the network to be end-to-end trainable. This enables us to freely learn the ConvPP rep-

Table 1: An experimental comparison on the OCR dataset by varying the number of training data. It demonstrates that the generalization error monotonically decreases when adding more data.

Training Data Percentage (%)	10	20	30	40	50	60	70	80	90
Generalization Error(%)	6.49	3.28	2.09	1.67	1.55	1.03	0.92	0.72	0.57

resentations on different scales of the ground-truth label space. The flexibility of the useful context that ConvPP can capture, now comes from two aspects: (1) the convolution filter weights can be automatically learned during training. (2) the context range is also handled explicitly by multi-scale architecture design. One can imagine that ConvPP representations, learned on low-resolution ground-truth maps, are capable of modeling complex long range and high order semantic context, global object shape and spatial configuration, whereas representations learned on high-resolutions ground-truth maps are supposed to model local structures like local smoothness.

Given that we can learn $\mathbf{H_Y}$ from different scales, we are readily able to build the spatial correspondences between $\mathbf{H_X}$ and $\mathbf{H_Y^{(tr)}}$. One can concatenate the $\mathbf{H_Y}$ to any convolutional feature maps learned in the bottom-up CNN network, as long as they passed through the same number of downsampling layers.

Because our convolutional pseudoprior is learned directly from the ground-truth label space, and it does not condition on the input data at all, the choice of bottom-up CNN models are flexible. The complementary structural information provided by the ConvPP allows us to easily improve on state-of-the-art CNN architectures such as Fully Convolutional Neural Networks (FCN).
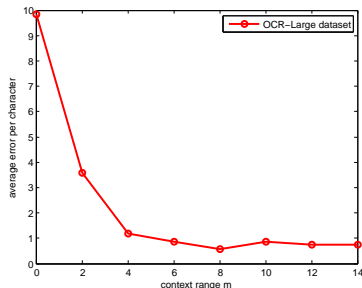
4   Experiments

In this section, we show experimental results on four benchmark datasets across different domains, namely FAQ (Natural language processing), OCR (Sequential image recognition), Pascal-Context (Semantic segmentation) and SIFT Flow (Scene labeling).
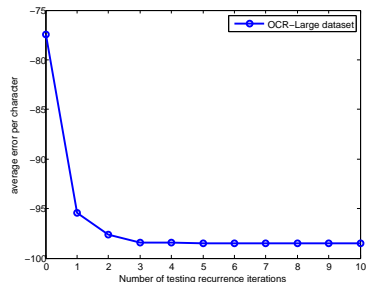
4.1   Sequential Labeling: 1-D case

First, we explore the effectiveness of our proposed framework on two 1-D structured (sequential) labeling tasks: handwritten OCR [6] and FAQ sentence labeling [34]. In these two 1-D toy examples, the pseudoprior model is implemented as a fully connected layer whose inputs are one-hot-encoding of neighboring labels (excluding the to-be-predicted label itself). When we slide the model over the input sequences, this layer can be viewed as a 1D convolutional layer with kernel size m (m is the context window size) and hole size 1.

Handwritten OCR This dataset contains $6,877$ handwritten words, corresponding to various writings of 55 unique words. Each word is represented as a

series of handwritten characters; there are $52,152$ total characters. Each character is a binary 16×8 image, leading to 128-dimensional binary feature vectors. Each character is one of the 26 letters in the English alphabet. The task is to predict the identity of each character. We first resize all the OCR characters to the same size (28×28) and build a standard 5-layer LeNet [35]. The label context part has a single-hidden-layer MLP with 100 units. We normalize each image to zero-mean and unit-variance.

Fig. 4: (a) comparison of the generalization error on the OCR handwritten dataset by varying the context window length. (b) the generalization error on the OCR handwritten dataset as the number of testing iterations varies.

FAQ The FAQ dataset consists of 48 files collecting questions and answers gathered from 7 multi-part UseNet FAQs. There are a total of $55,480$ sentences across the 48 files. Each sentence is represented with 24-dimensional binary feature vector. [34] provides a description of the features). We extended the feature set with all pairwise products of the original 24 features, leading to a 600-dimensional feature representation. Each sentence in the FAQ dataset is given one of four labels: (1) head, (2) question, (3) answer, or (4) tail. The task is to predict the label for each sentence. We train a 3-hidden layer fully-connected network with [32, 64, 128] hidden units respectively. A single-hidden-layer MLP with 100 hidden units is trained on ground-truth labels.

For both of the dataset, two hyper-parameters are specified by cross-validation: we set the context window size to be 7 (for OCR) and 5 (for FAQ); the number of iterations during testing to be 10.

Results. The results in Table 2 and Table 3 show that our proposed framework effectively models 1-D sequential structures and achieves better results for structured labeling as compared to previous methods. Several interesting observations: (1) on OCR dataset, compared to a kernel methods with hand-crafted features, our deep hybrid model performs worse on smaller dataset. But our deep learning approach performs better when the amount of training data increases. That is also the reason why ConvPP framework is important: handcrafted fea-

Table 2: Performance (error rate (%)) of structured labeling methods on the OCR dataset.

Methods	small	large
Linear-chain CRF [36]	21.62	14.20
M^{3}N [36]	21.13	13.46
SEARN [10]	-	9.09
SVM + CRF ([37])	-	5.76
Neural CRF [36]	10.8	4.44
Hidden-unit CRF [38]	18.36	1.99
Fixed-point [32]	**2.13**	0.89
NN without ConvPP	15.73	3.69
ConvPP (ours)	6.49	**0.57**

Table 3: Performance (error rate (%)) of structured labeling methods on the FAQ sentence labeling dataset.

Methods	error
Linear SVM [36]	9.87
Linear CRF [38]	6.54
NeuroCRFs [36]	6.05
Hidden-unit CRF [38]	4.43
NN without ConvPP	5.25
ConvPP (ours)	**1.09**

tures and kernel methods are hard to be applied to many high-level vision tasks where big data is available. (2) ConvPP context window length reflects the range of context needed, we can see from Figure 4 (a) that the generalization error converges when the context window length is about 7, which is the typical length of a word in the dataset. (3) Figure 4 (b) shows that though we set the max number of testing iterations to be 10, with only 3 to 4 iterations at test time, the generalization error converges. That shows that the inference of our ConvPP model can be efficient. (4) The experiment in the simple sentence classification task shows that ConvPP has the potential to be applied on more NLP tasks such as sequence modeling. (5) To show the effectiveness of the proposed approach, especially the convolutional pseudoprior part, we also perform the ablation study where we train a bottom-up network with exactly the same parameter settings. From the results we can see that without the structural information learned from the output label space, the performance decreases a lot.

4.2   Image Semantic Labeling: 2-D case

We then focus on two more challenging image labeling tasks: semantic segmentation and scene labeling. Most of deep structured labeling approaches evaluate their performance on the popular Pascal VOC object segmentation dataset [39], which contains only 20 object categories. Recently, CNN based methods, notably built on top of FCN [20], succeeded and dominated the Pascal-VOC leader-board where the performance (mean I/U) saturated to around 80%. Here we instead evaluate our models on the much more challenging Pascal-Context dataset [40], which has 60 object/stuff categories and is considered as a fully labeled dataset (with much fewer pixels labeled as background). We believe the top-down contextual information should play a more crucial role in this case. We also evaluated our algorithm on SIFT Flow dataset [41] to evaluate our algorithm on the task of traditional scene labeling. In both experiments, the performance is measured by the standard mean intersection-over-union (mean I/U).
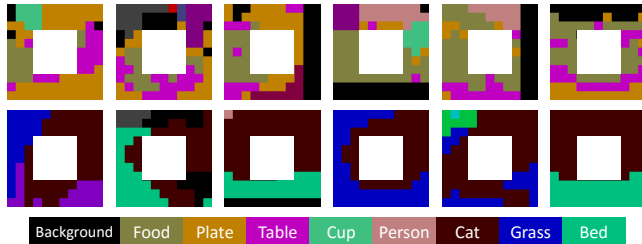
Fig. 5: Visualization of 2 filters. Each row displays top 6 label patches that produce highest activation for a filter.

Multi-scale integration with FCN. We build our hybrid model using FCN as the bottom-up CNN, and directly use the pre-trained models provided by the authors. FCN naturally handles multi-scale predictions by upgrading its 32-stride (32s) model to 16-stride (16s)/8-stride (8s) variants, where the final labeling decisions are made based on both high-level and low-lever representations of the network. For the 32s model and an input image size 384×512, the size of the final output of FCN, after 5 pooling layers, is 12×16. As discussed in our formulation, ConvPP can be integrated into FCN by downsampling the ground-truth maps accordingly.

Hyper-parameter settings. For all the 2-D labeling experiments, the number of channels in the donut filter convolution layer is 128. Adding more filters does not improve the performance. We keep all the hyper-parameters for the original bottom-up FCN network. The learning rates for the top-down ConvPP network are set to be 1e-7 for 32s, 1e-8 for 16s and 1e-9 for 8s variant. We choose the kernel size k of our donut filters by cross validation. The size of the hole in the middle is set to $\lfloor k/2 \rfloor \times \lfloor k/2 \rfloor$. In following two image labeling experiments we evaluate two configurations of donut filter, namely donut filter with small (7×7) kernel size, and large (11×11) kernel size. The comparison of results for those two configurations is shown in Table 4. The choice of donut filter size is crucial to the pseudoprior representation learning.

Sparse donut filter. We use 11×11 donut filters with 5×5 holes for Pascal-Context dataset since it achieves the best performance. The kernel covers a large portion of the 32-stride downsampled ground-truth map, and is therefore able to capture long range context. However, these large filters are typically very hard to learn. Inspired by [42], we reduce the number of learnable parameters while

Table 4: Comparison of the results by varying the donut filter kernel size.

dataset	kernel size	mean IU
PASCAL-Context	7×7	40.3
	11×11	**41.0**
SIFT Flow	**7×7**	**40.7**
	11×11	32.4

keeping the context range. Starting from a randomly initialized 6x6 kernel, we dilate the convolution kernel by inserting zeros between every neighboring position. Zero-valued locations are fixed to zero throughout the training. The resulting kernel is of size 11x11 but only 6x6 parameters are learnable.

Training and testing process. We follow the procedure of FCN to train our multi-scale hybrid model by stages. We train the ConvPP-32s model first, then upgrade it to the ConvPP-16s model, and finally to the ConvPP-8s model. During testing, we found that 3 iterations are enough for our fixed-point approach to converge, thus we keep this parameter through out our experiments. One concern is if the iterative testing process could diverge. Interestingly, in all our experiments (1-D and 2-D), the results are improved monotonically and converged. This shows that the pseudoprior learning process is stable and the fixed-point solver is effective. The input of ConvPP part is initialized with original FCN prediction since it is readily available.

Computational cost. Since we can utilize pretrained bottom-up network, training the single-layer top-down convolutional pseudoprior network is efficient. For Pascal-context dataset the training can be done in less than 1 hour on a single Tesla K40 GPU. The additional computational cost due to iterative inference procedure is also small. For 3 iterations of fixed-point inference, our ConvPP model only takes additional 150ms. Note that all previous works using CRFs (either online or offline) also require testing-stage iterative process.

Pascal-Context. This dataset contains ground truth segmentations fully annotated with 60-category labels (including background), providing rich contextual information to be explored. We follow the standard training + validation split as in [40,20], resulting in 4,998 training images and 5,105 validation images.

Table 5 shows the performance of our proposed structured labeling approach compared with FCN baselines and other state-of-the-art models.

We hope to evaluate our approach in a way that allows fair comparison with FCN, which does not explicitly handle structural information. Therefore we carefully control our experimental settings as follows: **(1)** We do not train the bottom-up CNN models for all the experiments in Training Stage 1, and use the pre-trained models provided by the authors. **(2)** We train the top-down ConvPP network

Table 5: Results on Pascal-Context dataset [40]. ConvPP outperforms FCN baselines and previous state-of-the-art models. † is trained with additional data from COCO.

	mean IU
O_2P [43]	18.1
CFM (VGG+SS) [44]	31.5
CFM (VGG+MCG) [44]	34.4
CRF-RNN [21]	39.3
BoxSup† [45]	40.5
FCN-32s [20]	35.1
ConvPP-32s (ours)	37.1
FCN-16s [20]	37.6
ConvPP-16s (ours)	40.3
FCN-8s [20]	37.8
ConvPP-8s (ours)	**41.0**

(a) image   (b) ground-truth   (c) FCN-8s   (d) ConvPP-8s-Iter1   (e) ConvPP-8s-Iter2   (f) ConvPP-8s-Iter3
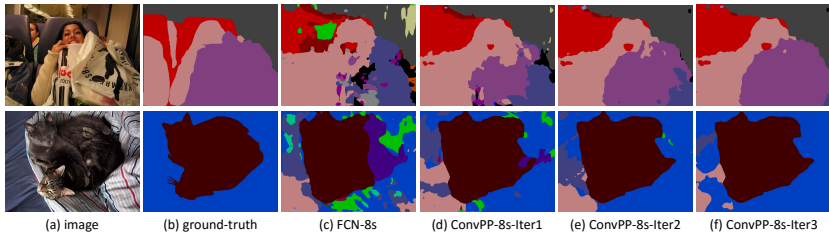
Fig. 6: Iterative update of labeling results during testing. Segmentation results are gradually refined.

(Training stage 2) independently on each scale, namely 32s, 16s and 8s. **(3)** To train the hybrid models at a certain scale, we only use the pre-trained FCN models at the corresponding scale. (ConvPP-32s can only use the FCN-32s representations.) **(4)** For all the experiments, we fix the learning rate of the FCN part of the hybrid model, namely all the convolutional layers from Conv1_1 to fc7, to be zero. The reason we freeze the learning rate of FCN is to do an **ablation study**: we want to show the performance gain incorporating the ConvPP part. Intuitively context information should help more in high-level structural prediction rather than improving low-level appearance features. The experiment results support this claim: we get 40.89 when joint-tuning the parameters in the bottom-up FCN-8s network, the difference is negligible. Our methods consistently outperform the FCN baselines. We show the results for ConvPP 32s (structural information integrated in layer pool 5), ConvPP 16s (pool5+pool4) and ConvPP 8s (pool5 + pool4 + pool3) to analyze the effect of multi-scale context learning. The results have been consistently improved by combining finer scales.

Our method also outperforms other state-of-the-art models built on FCN, notably CRF-RNN [21], which also explicitly handles structured labeling problem by integrating a fully-connected CRF model into the FCN framework; and BoxSup [45], which is trained with additional COCO data. This clearly shows the effectiveness of our ConvPP model in capturing complex inter-dependencies in structured output space. Example qualitative results of our ConvPP-8s compared to baseline FCN-8s model can be found in the supplementary material. Figure 6 shows how our labeling results are iteratively refined at inference time. With multi-scale architecture design, our method leverages both short- and long-range context to assist the labeling task. ConvPP is able to recover correct labels as well as suppress erroneous label predictions based on contextual information. In Figure 5, we visualize 2 learned donut filters on 32-stride ground-truth maps by displaying label patches that produce top 6 activations (as done in [46]). It is shown that our filters are learned to detect complex label context patterns, such as "food-plate-tabel", "cat on the bed" and "cat on the grass".

SIFT Flow We also evaluate our method on scene labeling task, where context is also important in accurate labeling. SIFT Flow dataset [41] contains 2,688 images with 33 semantic categories. A particular challenge for our ConvPP model

for this dataset is the relatively small image/ground-truth map size (256×256), which means the 32-stride output is only 8×8. Downsampling the ground-truth map to this scale could potentially lead to loss in useful context information. In addition, the finest model provided by [20] is FCN-16s instead of FCN-8s.

To alleviate this problem, we train our own FCN-8s model (pre-trained with the provided FCN-16s model) as our baseline and build our ConvPP-8s on top of it. Also because of the size of the image in the dataset, as shown in Table 4, 11×11 donut filters perform poorly. Thus we choose the donut filters with kernel size 7×7 and hole size 3×3, and the sparsification operation is not needed. The testing procedure is the same as that of Pascal-Context dataset.

According to Table 6, our ConvPP models consistently outperform corresponding FCN baselines. The improvement of ConvPP-16s

Table 6: Results on SIFT Flow dataset [41]. Our methods outperform the strong FCN baselines. Improvement of ConvPP-8s vs FCN-8s is more significant than that of ConvPP-16s vs FCN-16s, since higher resolution ground truth map carries more structured information.

	mean IU
FCN-16s [20]	39.1
ConvPP-16s (ours)	39.7
FCN-8s [20]	39.5
ConvPP-8s (ours)	**40.7**

model is relatively small, which might result from the limited resolution of ground-truth maps (256 x 256). With higher ground-truth resolution, ConvPP-8s outperforms the stronger FCN-8s baseline by 1.2% in mean I/U. This substantiate that our proposed pseudoprior learning framework is effective in learning structural information from the ground-truth labeling space.

5   Conclusions

We propose a new method for structured labeling by developing convolutional pseudoprior (ConvPP) on the ground-truth labels. ConvPP learns convolution in the labeling space with improved modeling capability and less manual specification. The automatically learns rich convolutional kernels can capture both short- and long- range contexts combined with a multi-scale hybrid model architecture design. We use a novel fixed-point network structure that facilitates an end-to-end learning process. Results on structured labeling tasks across different domains shows the effectiveness of our method.

References

1. Elman, J.L.: Finding structure in time. Cognitive Science **14**(2) (1990) 179–211
2. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional Random Fields. In: ICML. (2001)
3. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV. (2006)
4. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE PAMI **6**(6) (1984) 721–741
5. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large Margin Methods for Structured and Interdependent Output Variables. JMLR **6** (2005) 1453–1484
6. Taskar, B., Guestrin, C., Koller, D.: Max-Margin Markov Networks. In: NIPS. (2003)
7. Finley, T., Joachims, T.: Training structural SVMs when exact inference is intractable. In: ICML. (2008)
8. Tu, Z.: Auto-Context and Its Application to High-Level Vision Tasks. In: CVPR. (2008)
9. Heitz, G., Gould, S., Saxena, A., Koller, D.: Cascaded Classification Models. In: NIPS. (2008)
10. Daumé, H.I., Langford, J., Marcu, D.: Search-based Structured Prediction. Machine Learning (2009)
11. Wolpert, D.H.: Stacked Generalization. Neural networks **5**(2) (1992) 241–259
12. Ames Jr, A.: Visual perception and the rotating trapezoidal window. Psychological Monographs: General and Applied **65**(7) (1951) i
13. Marr, D.: Vision: A computational approach (1982)
14. Gibson, J.J.: A theory of direct visual perception. Vision and Mind: selected readings in the philosophy of perception (2002) 77–90
15. Kersten, D., Mamassian, P., Yuille, A.: Object perception as Bayesian inference. Annu. Rev. Psychol. **55** (2004) 271–304
16. Tu, Z., Chen, X., Yuille, A.L., Zhu, S.C.: Image parsing: Unifying segmentation, detection, and recognition. IJCV **63**(2) (2005) 113–140
17. Borenstein, E., Ullman, S.: Combined top-down/bottom-up segmentation. IEEE PAMI **30**(12) (2008) 2109–2125
18. Wu, T., Zhu, S.C.: A numerical study of the bottom-up and top-down inference processes in and-or graphs. IJCV **93**(2) (2011) 226–252
19. Krahenbuhl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS. (2011)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. (2015)
21. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. arXiv preprint arXiv:1502.03240 (2015)
22. Lin, G., Shen, C., Reid, I., Hengel, A.v.d.: Deeply learning the messages in message passing inference. In: NIPS. (2015)
23. Besag, J.: Efficiency of pseudolikelihood estimation for simple gaussian fields. Biometrika (1977) 616–618
24. Zhu, S.C., Mumford, D.: A Stochastic Grammar of Images. Foundations and Trends in Computer Graphics and Vision **2**(4) (2006) 259–362

25. He, X., Zemel, R.S., Carreira-Perpiñán, M.Á.: Multiscale conditional random fields for image labeling. In: CVPR. (2004)
26. Kae, A., Sohn, K., Lee, H., Learned-Miller, E.: Augmenting crfs with boltzmann machine shape priors for image labeling. In: CVPR. (2013)
27. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
28. Hinton, G., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation (2006)
29. Snoek, J., Adams, R.P., Larochelle, H.: Nonparametric Guidance of Autoencoder Representations using Label Information. JMLR (2012)
30. Bengio, Y., Thibodeau-Laufer, E., Alain, G., Yosinski, J.: Deep generative stochastic networks trainable by backprop. arXiv preprint arXiv:1306.1091 (2013)
31. Tu, Z., Narr, K.L., Dollár, P., Dinov, I., Thompson, P.M., Toga, A.W.: Brain anatomical structure segmentation by hybrid discriminative/generative models. IEEE Tran. on Medical Imaging **27**(4) (2008) 495–508
32. Li, Q., Wang, J., Wipf, D., Tu, Z.: Fixed-Point Model for Structured Labeling. In: ICML. (2013) 214–221
33. Makhzani, A., Frey, B.: Winner-Take-All Autoencoders. In: NIPS. (2015)
34. McCallum, A., Freitag, D., Pereira, F.C.: Maximum Entropy Markov Models for Information Extraction and Segmentation. In: ICML. (2000)
35. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. (1998)
36. Do, T., Arti, T.: Neural conditional random fields. In: AISTATS. (2010)
37. Hoefel, G., Elkan, C.: Learning a two-stage SVM/CRF sequence classifier. In: CIKM, ACM (2008)
38. van der Maaten, L., Welling, M., Saul, L.K.: Hidden-unit conditional random fields. In: AISTATS. (2011)
39. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (2012)
40. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., et al.: The role of context for object detection and semantic segmentation in the wild. In: CVPR. (2014)
41. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: CVPR. (2009)
42. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In: ICLR. (2015)
43. Carreira, J., Caseiro, R., Batista, J., Sminchisescu, C.: Semantic segmentation with second-order pooling. In: ECCV. (2012)
44. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: CVPR. (2015)
45. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: ICCV. (2015)
46. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Computer Vision–ECCV 2014. Springer (2014) 818–833

Supplementary Material for Paper:
Top-Down Learning for Structured Labeling with Convolutional Pseudoprior

(a) Image   (b) ground-truth   (c) FCN-8s   (d) ConvPP-8s