Supplementary Material for
Towards a Unified Compositional Model for Visual Pattern Modeling

Wei Tang, Pei Yu, Jiahuan Zhou and Ying Wu
Northwestern University
2145 Sheridan Road, Evanston, IL 60208
{wtt450, pyi980, jzt011, yingwu}@eecs.northwestern.edu

1. Back-propagation (BP)

We derive the BP formulations for the And-layer[1]. Recall its definition is

$$S_u^*(w_u) = \sum_{v \in \mathcal{V}^{l(u)-1}} \max_{\bar{w}_v \in \mathbb{D}_v} \lambda_{u,v}^{and}(\bar{w}_v) + \lambda_{u,v}^{con} S_v^*(w_v) \quad (1)$$

After each forward pass, the objective function J is computed. Let $H_{u,w_u}(v, \bar{w}_v) \in \{0, 1\}$ indicate whether $\bar{w}_v$ is the location with the maximum score in Eq. (1) for the output unit (u, w_u) and input channel v. It should satisfy: $\sum_{\bar{w}_v \in \mathbb{D}_v} H_{u,w_u}(v, \bar{w}_v) = 1$. Then, the partial derivatives of J w.r.t. the input units and parameters can be obtained as follows:

$$\frac{\partial J}{\partial S_v^*(w_v)} = \sum_{\substack{u \in \mathcal{V}^{l(v)+1} \\ \bar{w}_v \in \mathbb{D}_v}} \frac{\partial J}{\partial S_u^*(w_u)} H_{u,w_u}(v, \bar{w}_v) \lambda_{u,v}^{con} \quad (2)$$

$$\frac{\partial J}{\partial \lambda_{u,v}^{con}} = \sum_{w_u} \frac{\partial J}{\partial S_u^*(w_u)} \sum_{\bar{w}_v \in \mathbb{D}_v} H_{u,w_u}(v, \bar{w}_v) S_v^*(w_v) \quad (3)$$

$$\frac{\partial J}{\partial \lambda_{u,v}^{and}(\bar{w}_v)} = \sum_{w_u} \frac{\partial J}{\partial S_u^*(w_u)} H_{u,w_u}(v, \bar{w}_v) \quad (4)$$

It should be clear that $\bar{w}_v$, w_v and w_u are related by $\bar{w}_v \equiv w_v - w_u$: knowing two of them will determine the other.

2. Extended model

As mentioned on Page 3 of the paper, by including extra data potentials, we can associate the higher-level And-nodes with the observations. Then, the score function of the AOG becomes:

$$S(\Omega) = \sum_{u \in \mathcal{V}^{leaf}} \phi_u^{leaf}(w_u, \mathbf{I}) + \sum_{u \in \mathcal{V}^{or}} \phi_u^{or}(z_u, w_u) + \sum_{u \in \mathcal{V}^{and}} [\phi_u^{ext}(w_u, \mathbf{I}) + \sum_{v \in ch(u)} \phi_{u,v}^{and}(w_u, w_v)] \quad (5)$$

where $\phi_u^{ext}(w_u, \mathbf{I})$ denotes the data potentials for the And-nodes. It is defined similarly with the Leaf-node potentials:

$$\phi_u^{ext}(w_u, \mathbf{I}) = \lambda_u^{ext} \cdot f_u(w_u, \mathbf{I}; \Theta) \quad (6)$$

where λ_u^{ext} is the part classifier/filter; $f_u(w_u, \mathbf{I}; \Theta)$ is CNN features extracted at location w_u of image $\mathbf{I}$ with Θ being the collection of the CNN's parameters.

Following the derivations in the paper, the extended And-node model is:

$$S_u(\Omega_u) = \phi_u^{ext}(w_u, \mathbf{I}) + \sum_{v \in \mathcal{V}^{l(u)-1}} [\lambda_{u,v}^{and}(\bar{w}_v) + \lambda_{u,v}^{con} S_v(\Omega_v)] \quad (7)$$

Correspondingly, the And-layer becomes

$$S_u^*(w_u) = \phi_u^{ext}(w_u, \mathbf{I}) + \sum_{v \in \mathcal{V}^{l(u)-1}} \max_{\bar{w}_v \in \mathbb{D}_v} [\lambda_{u,v}^{and}(\bar{w}_v) + \lambda_{u,v}^{con} S_v^*(w_v)] \quad (8)$$

Thus, the extended And-layer is the summation of the basic And-layer in Eq. (1) and the CNN activations.

3. Computational complexity and running time

Assume H_{in}, W_{in} and C_{in} denote the height, width and channel number of the input feature maps, respectively; H_{out}, W_{out} and C_{out} denote the height, width and channel number of the output feature maps, respectively; K is the side length of a square kernel. The forward pass of an And-layer, defined in Eq. (1), is of computational complexity $O(K^2 C_{in} H_{out} W_{out} C_{out})$. For the backward pass, the complexities of Eqs. (2)-(4) are the same[2]: $O(C_{in} H_{out} W_{out} C_{out})$. In comparison, the complexities of the forward and backward passes of a convolution layer are both $O(K^2 C_{in} H_{out} W_{out} C_{out})$. The computational complexity of an Or-layer is the same with that

[1] The Primitive-layer is a CNN. The Or-layer is similar with the max-pooling layer and Maxout layer [3]. So, we omit their BP derivations here.

[2] In our implementation of the backward pass, coordinates of the selected locations $\bar{w}_v$, instead of the binary indicators $H_{u,w_u}(v, \bar{w}_v)$, are stored to reduce computation and memory costs.

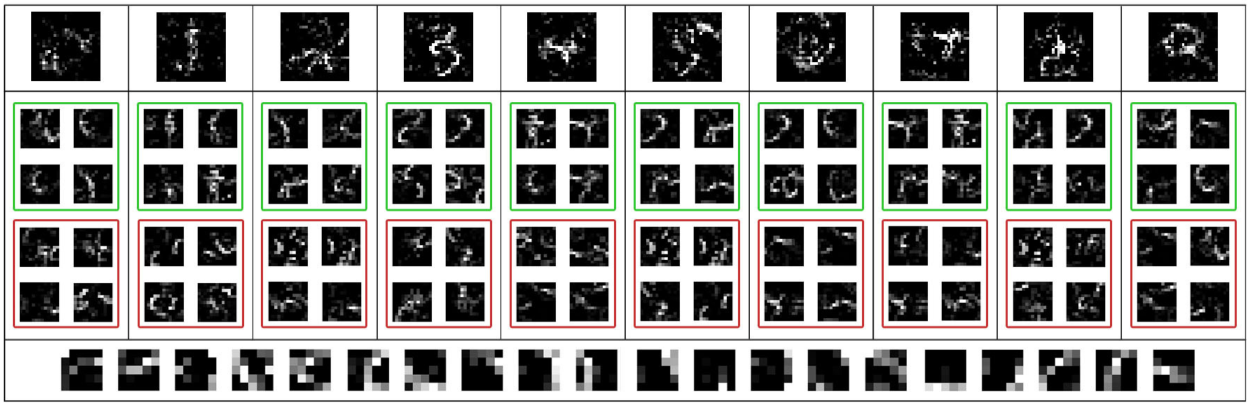

Figure 1. Node filters learned on the MNIST dataset. **Only positive values of each filter are displayed.** Each higher-level filter is composed from the child filters using the learned parameters. Bottom row: 20 primitive filters. Top row: 10 Object-level filters. Middle row: 80 Or-node filters. The 4 filters in a green (red) box are those most relevant (irrelevant) to the corresponding object filter, evaluated by λ_{uv}^{con}. Due to part sharing, some filters may occur in multiple columns.

of a Maxout layer: $O(C_{in}H_{in}W_{in})$ for forward pass and $O(C_{out}H_{out}W_{out})$ for backward pass. For an architecture with multiple layers, its computational complexity is just the summation of the complexities of each layer. Compared with our CompNet, the previous compositional models [5, 12, 10, 13, 14, 9] are less scalable. In the forward pass (inference), they search the parts in the whole image space instead of a local window. This leads to a complexity $O(C_{in}H^2W^2C_{out})$ for each composition level, where H and W denote the image size, C_{in} and C_{out} denote the child number of each parent and the parent number. Note H and W are far larger than K. To estimate the parameters, latent structural learning, based on the concave-convex procedure (CCCP) [11], is usually exploited. Each iteration contains two steps: i) Given the current parameter estimation, infer the optimal values of the latent variables. This is of the same complexity with the forward pass; ii) Solve a standard structural SVM problem [6] without latent variables. The second step is an iterative process and computationally expensive for large training set and deep structures [12]. In each sub-iteration, all latent states of all training examples have to be checked to create/update a working set [12]. Then, a SVM dual problem over this working set is optimized via the Sequential Minimal Optimization (SMO) [4].

Since our CompNet is trained via a GPU[3] while previous compositional models are learned using CPUs, it is unfair to compare their running times directly. For example, it takes the 3-layer model [12] 25 hours to train one object class on the PASCAL VOC 2007 dataset; The training of our deeper CompNet, as described in the paper, for all the 20 object classes takes less than 4 hours. Thus, we compare the CompNet with its CNN and Maxout Network counterparts. Specifically, in the ICDAR-03 character recognition task, all the three networks have the same depth, width and batch size, *i.e*. 128, as described in the paper. Using the same GPU, they achieve the same efficiency: training 100 iterations costs 8 seconds and testing on all the 5400 samples costs 1 second. This agrees with our complexity analysis.

4. Experiments

Fig. 1 here is the same with Fig. 6 in the paper except that only positive parts of the composed filters are displayed. We include it here for assistant visualization.

Tab. 1 supplements the experiment of object detection on the PASCAL VOC 2007 dataset with the average precisions (APs) of each method on the 20 object classes. We also include the detection results obtained on the PASCAL VOC 2012 dataset in Tab. 2. We can observe that the proposed CompNet outperforms the baseline methods on both datasets. However, the proposal-based approaches, *i.e*., CompNets and R-CNNs, fail to detect bottles well. This is due to the bad localization of this object class by the adopted object proposal method [7].

5. Discussion

For structure modeling, we explicitly parametrize and learn the connections. As we observed in our experiments, the learned connections are generally sparse. For example, Fig. 2 shows the histogram of the learned connection parameters of the CompNet in the MNIST experiment. Since most connections have strengths close to 0 and can be omitted, the learned model corresponds to an AOG with sparse connections.

In the experiments, we have tested both And-Or-And and

[3] NVIDIA TITAN X with 12 GB memory

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
CNN-DPM [2]	44.6	65.3	32.7	24.7	35.1	54.3	56.5	40.4	26.3	49.4	43.2	41	61	55.7	**53.7**	25.5	47	39.8	47.9	**59.2**	45.2
CNN-DPM [8]	49.3	**69.5**	31.9	28.7	40.4	61.5	61.5	41.5	25.5	44.5	47.8	32	67.5	61.8	46.7	25.9	40.5	46	57.1	58.2	46.9
RCM [12]	29.4	55.8	14.3	28.6	44	51.3	38.4	36.8	9.4	21.3	19.3	12.5	50.4	19.7	36.6	15.1	20	25.2	25.1	39.3	29.6
R-CNN p5 [1]	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	**26.7**	55.5	43.4	43.1	57.7	59	45.8	**28.1**	**50.8**	40.6	53.1	56.4	47.3
AOT [5]	35.3	60.2	16.6	29.5	**53**	57.1	49.9	48.5	11.0	23.0	27.7	13.1	58.9	22.4	41.4	16.0	22.9	28.6	37.2	42.4	34.7
Ours	**61.5**	62.3	**47.3**	**37.6**	17.9	**62.8**	**67.3**	**68.8**	23.3	**55.7**	**53.8**	**57.5**	**68.4**	**62.3**	53.6	22.3	47	**53.1**	**64.4**	52.4	**52.0**

Table 1. Detection average precision (in %) on Pascal VOC 2007.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
CNN-DPM [2]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	41.6
CNN-DPM [8]	63.3	**60.2**	33.4	24.4	**33.6**	60	44.7	49.3	19.4	36.6	30.2	40.7	57.7	61.4	52.3	**21.2**	44.4	37.9	51.1	**52.2**	43.7
Ours	**70.1**	**60.2**	**47.5**	**29.9**	20.4	**61.5**	**55.0**	**75.0**	**22.4**	**50.0**	**34.9**	**68.4**	**64.5**	**67.8**	**53.9**	20.8	**50.2**	**44.8**	**58.7**	48.8	**50.3**

Table 2. Detection average precision (in %) on Pascal VOC 2012.

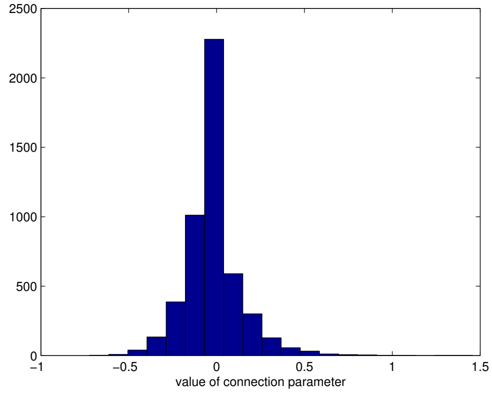

Figure 2. Histogram of the learned connection parameters of the CompNet in the MNIST experiment.

And-Or-And-Or architecture patterns. Intuitively, the latter structure, which has one more Or-layer, can explicitly decouple the modes of patterns and increase the interpretability. However, we observed negligible differences between their practical performances in all the experiments. This can be explained for several reasons. (1) As discussed in Sec. 3.2 in the paper, our nonparametric And-node can model multimodal distributions. (2) For characters, there is no viewpoint or articulation variations. (3) For object detection, deep CNNs have captured some invariance to the pattern variations. We chose the And-Or-And structure since it is the simplest one without degradation in performance.

6. Codes and more visualization results

The codes and more visualization results can be found at the project web: http://www.ece.northwestern.edu/~wtt450/project/ICCV17_CompNet.html.

References

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

[2] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015. 3

[3] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. C. Courville, and Y. Bengio. Maxout networks. *ICML*, 2013. 1

[4] J. C. Platt et al. Using analytic qp and sparseness to speed training of support vector machines. *NIPS*, 1999. 2

[5] X. Song, T. Wu, Y. Jia, and S.-C. Zhu. Discriminatively trained and-or tree models for object detection. In *CVPR*, 2013. 2, 3

[6] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005. 2

[7] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2

[8] L. Wan, D. Eigen, and R. Fergus. End-to-end integration of a convolution network, deformable parts model and non-maximum suppression. In *CVPR*, 2015. 3

[9] J. Wang and A. L. Yuille. Semantic part segmentation using compositional model combining shape and appearance. In *CVPR*, 2015. 2

[10] T. Wu, B. Li, and S. C. Zhu. Learning and-or model to represent context and occlusion for car detection and viewpoint estimation. *PAMI*, 2016. 2

[11] A. L. Yuille, A. Rangarajan, and A. Yuille. The concave-convex procedure (cccp). *NIPS*, 2002. 2

[12] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 2, 3

[13] L. L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille. Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. In *CVPR*, 2010. 2

[14] L. L. Zhu, Y. Chen, and A. Yuille. Recursive compositional models for vision: Description and review of recent work. *Journal of Mathematical Imaging and Vision*, 2011. 2