Supplementary Materials

Anonymous ECCV submission

Paper ID 7

1 Training details

TDMPNet: For PASCAL3D+ and COCO, we first train prototypes individually for 10 epochs and then the whole network for 50 epochs; for MNIST, we first train prototypes individually for 10 epochs and then the whole network for 10 epochs. We use SGD with momemtum to train the model. Learning rate is 1e-3 for prototype training and 1e-4 for the whole network training. We use l_2 regularization for convolution layers. Batch normalization, data augmentation, or other regularization methods are not used.

The thresholds in Eq.(6) are dynamically determined: the upper threshold is the top 20% value among $\{a^4_{i,j}\}$, and the lower threshold is the top 80% value among $\{a^4_{i,j}\}$. The parameter γ in Eq.(2) is initialized as 20, and the hyper-parameter λ_1 and λ_2 in Eq.(10) are simply set to 1.

VGG-16: For PASCAL3D+, MNIST, and COCO, the last layer is first finetuned for 5 epochs and the whole network is finetuned for 10 epochs until convergence. The regularization and optimization method for it is the same as TDMPNet.

2 Parameter amount

The convolutional layers of TDMPNet are the same as convolutional layers of VGG. Let H, W, C denote the dimension of the output feature tensor, and N, M denote the category number and prototype number per category. The parameter amount of prototypes is $H \times W \times C \times N \times M$, which is equivalent to a fully connected layer with input size $H \times W \times C$ and output size $N \times M$. The parameter amount of feature dictionary is $N_D \times C$, where N_D is the component number of feature dictionary. As we discard fully connected layers in VGG, the parameters of TDMPNet is usually less than VGG. In the experiment settings, the parameter amount of TDMPNet is only 14 percent of VGG.

3 Analysis of prototype learning on different layers

As discussed in the experiment part, CompDictModel outperforms TDAPNet in LEVEL-3 'o' condition of PASCAL3D+. A difference between CompDictModel and TDAPNet is that TDAPNet learn prototypes from the pool-5 layer while CompDictModel learn compositional model from the pool-4 layer. We analyze

Table 1. Comparison of prototype learning on pool-4 layer and pool-5 layer for TDAPNet on PASCAL3D+. It shows that prototype learning on pool-5 layer outperforms pool-4 layer in zero or low occlusion conditions and when occlusion is white boxes, noise boxes, or textures, while prototype learning on pool-4 layer performs well when occlusion is object under high occlusion level.

PASCAL3D+ Classification under Occlusion

Occ. Area	0%	Level-1: 20-40%				Level-2: 40-60%				Level-3: 60-80%				Mean
Occ. Type	-	w	n	t	o	w	n	t	o	w	n	t	o	-
VGG	99.4	97.5	97.5	97.3	92.1	91.7	90.6	90.2	73.0	65.0	60.7	56.4	52.2	81.8
CompDictModel	98.3	96.8	95.9	96.2	94.4	91.2	91.8	91.3	91.4	71.6	80.7	77.3	87.2	89.5
pool-5, without recurrence	99.3	98.4	98.9	98.5	97.3	96.4	97.1	96.2	89.2	84.0	87.4	79.7	74.5	92.1
pool-5, with 1 recurrence	99.3	98.4	98.9	98.7	97.2	96.1	97.4	96.4	90.2	81.1	87.6	81.2	76.8	92.3
pool-4, without recurrence	98.1	97.6	98.0	97.8	96.4	94.8	96.3	95.2	92.0	76.5	85.2	79.5	84.2	91.7
pool-4, with 1 recurrence	98.0	97.3	97.9	97.8	96.0	94.6	96.1	95.3	91.7	77.5	85.9	80.5	84.4	91.8

the effect of prototype learning on different layers. As shown in Table 1, performance drops under most conditions while increases under object occlusion at level-2 and level-3 ’o’ conditions if we learn prototypes from the pool-4 layer. It performs similar with CompDictModel. A possible reason is that the information in the pool-4 layer is more local and part-based, so it does not perform as well as information from the pool-5 layer under zero or low occlusion conditions. But under severe occlusion with irregular shape, local information may play a more important role for object recognition, leading to its high performance under the certain condition. A mechanism to combine the information from the pool-4 layer and the pool-5 layer might utilize both of their advantages.

4 Visualization of different prototypes

As shown in Figure 1, four different prototypes in TDMPNet could mainly account for different spatial distribution caused by viewpoints or appearance.

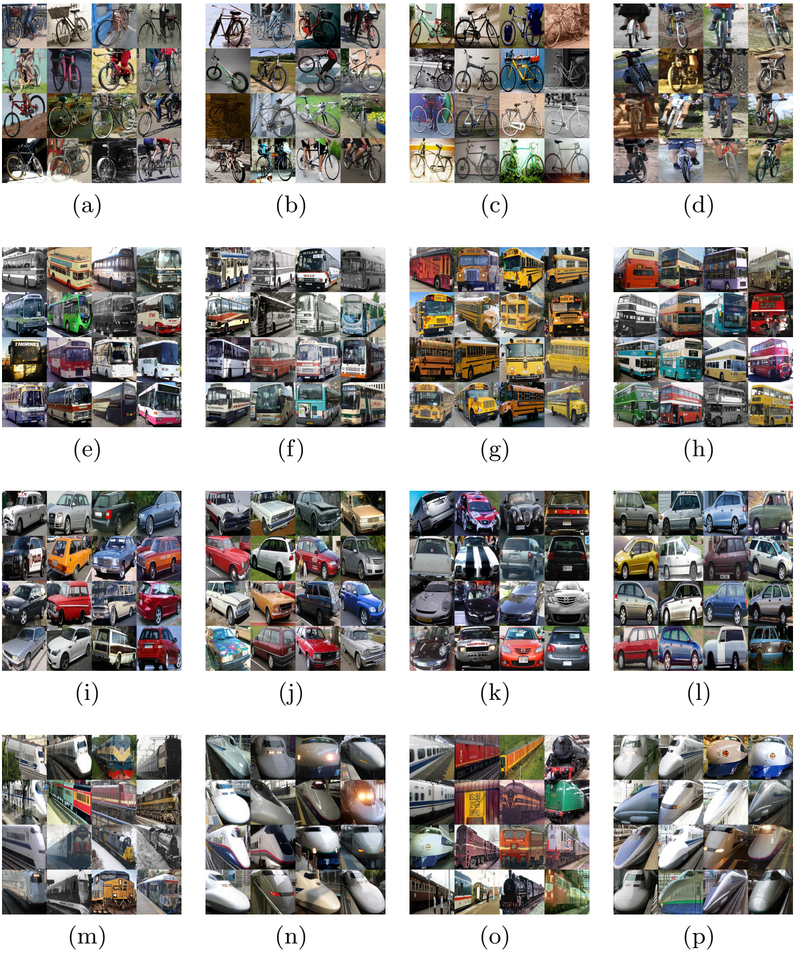

Fig. 1. Visualization of different prototypes. It shows that different prototypes could mainly account for different spatial distribution caused by viewpoints. And if there are more prototypes than viewpoints, some prototypes could focus on some specific features, like school bus in (g) and double-decker bus in (h), or different appearance shown in (m), (n), (o), and (p)