CNN-RNN: A Unified Framework for Multi-label Image Classification

Jiang Wang Yi Yang Junhua Mao Zhiheng Huang Chang Huang
Wei Xu

CVPR 2016

Contents

Motivation

Sacrifice computational complexity to model more complicated label relationships

Airplane | Great Pyrenees | Archery

Sky, Grass, Runway | *Dog, Person, Room* | *Person, Hat, Nike*

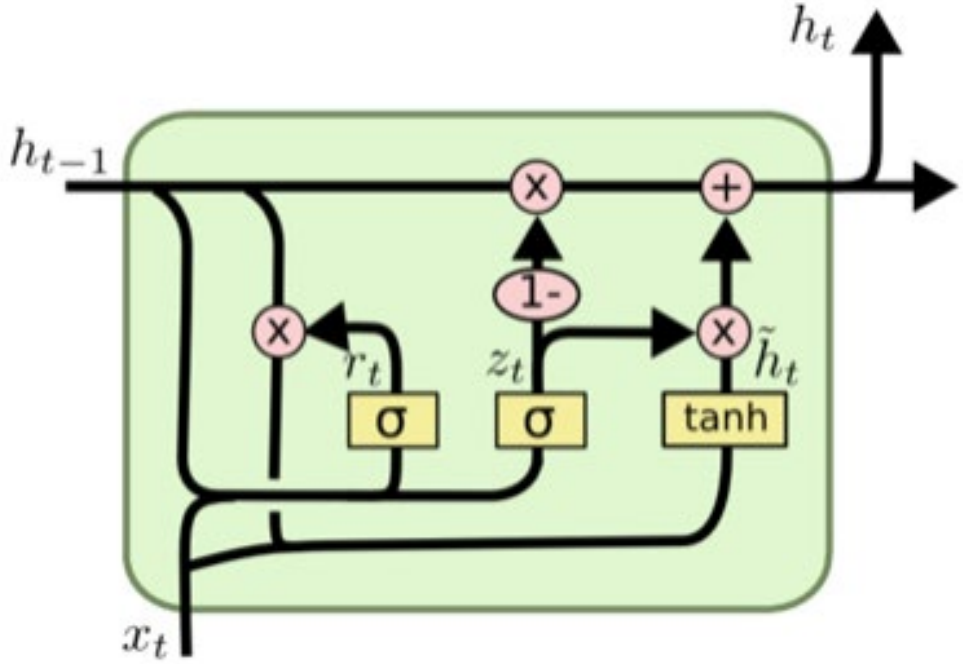
Figure 1. We show three images randomly selected from ImageNet 2012 classification dataset. The second row shows their corresponding label annotations. For each image, there is only one label (*i.e.* Airplane, Great Pyrenees, Archery) annotated in the ImageNet dataset. However, every image actually contains *multiple labels*, as suggested in the third row.

Small objects are hard to recognize by itself

RNN:LSTM

RNN with LSTM can effectively model the long-term temporal dependency in a sequence

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$



Methods

Extract Features:

Objective to guide:

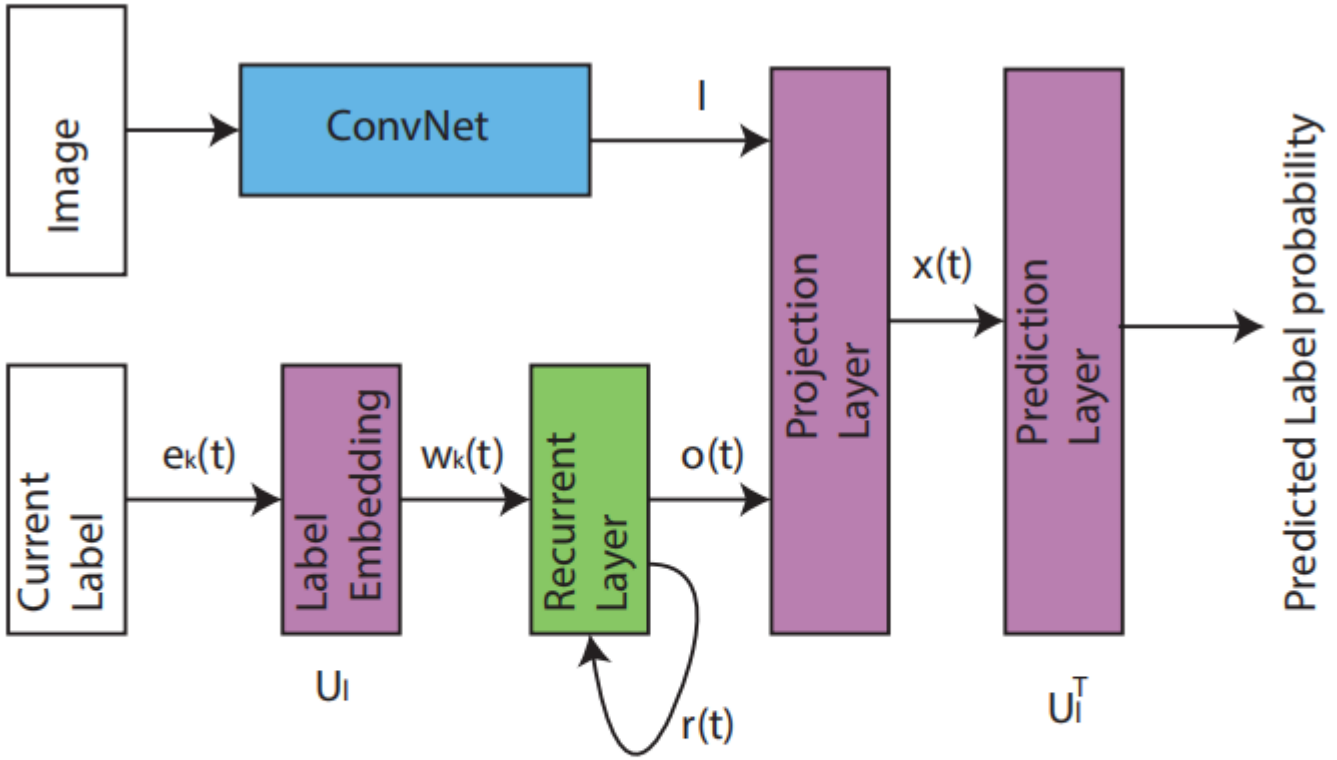


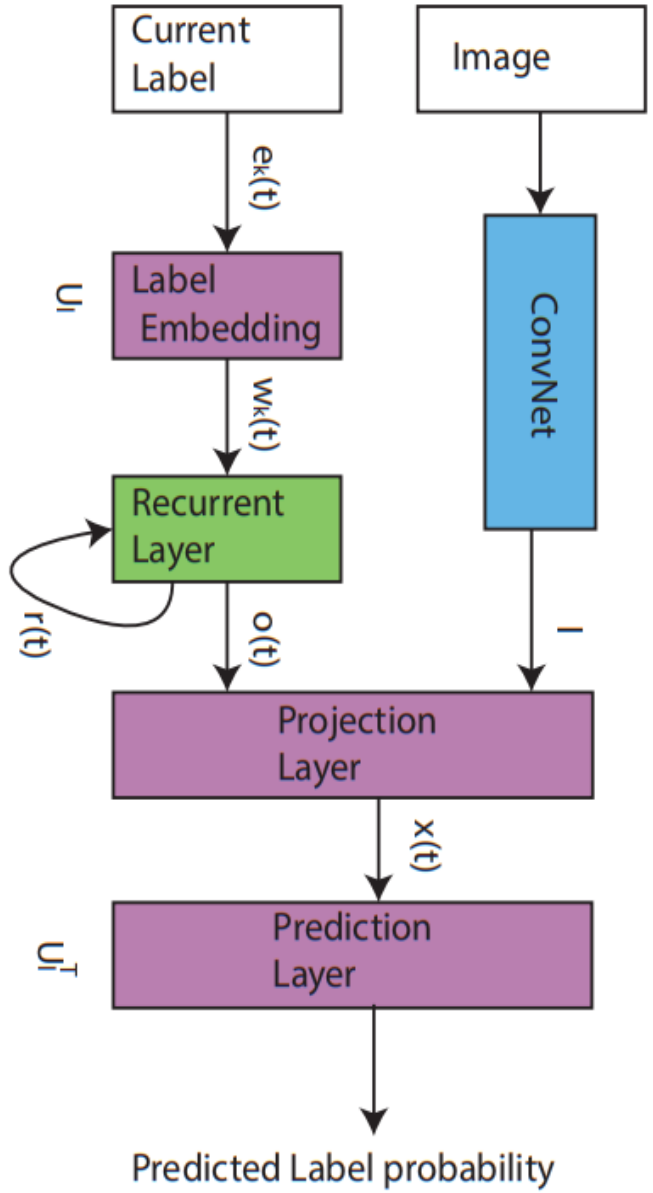
Figure 4. The architecture of the proposed RNN model for multi-label classification. The convolutional neural network is employed as the image representation, and the recurrent layer captures the information of the previously predicted labels. The output label probability is computed according to the image representation and the output of the recurrent layer.

$$w_k = U_l.e_k.$$

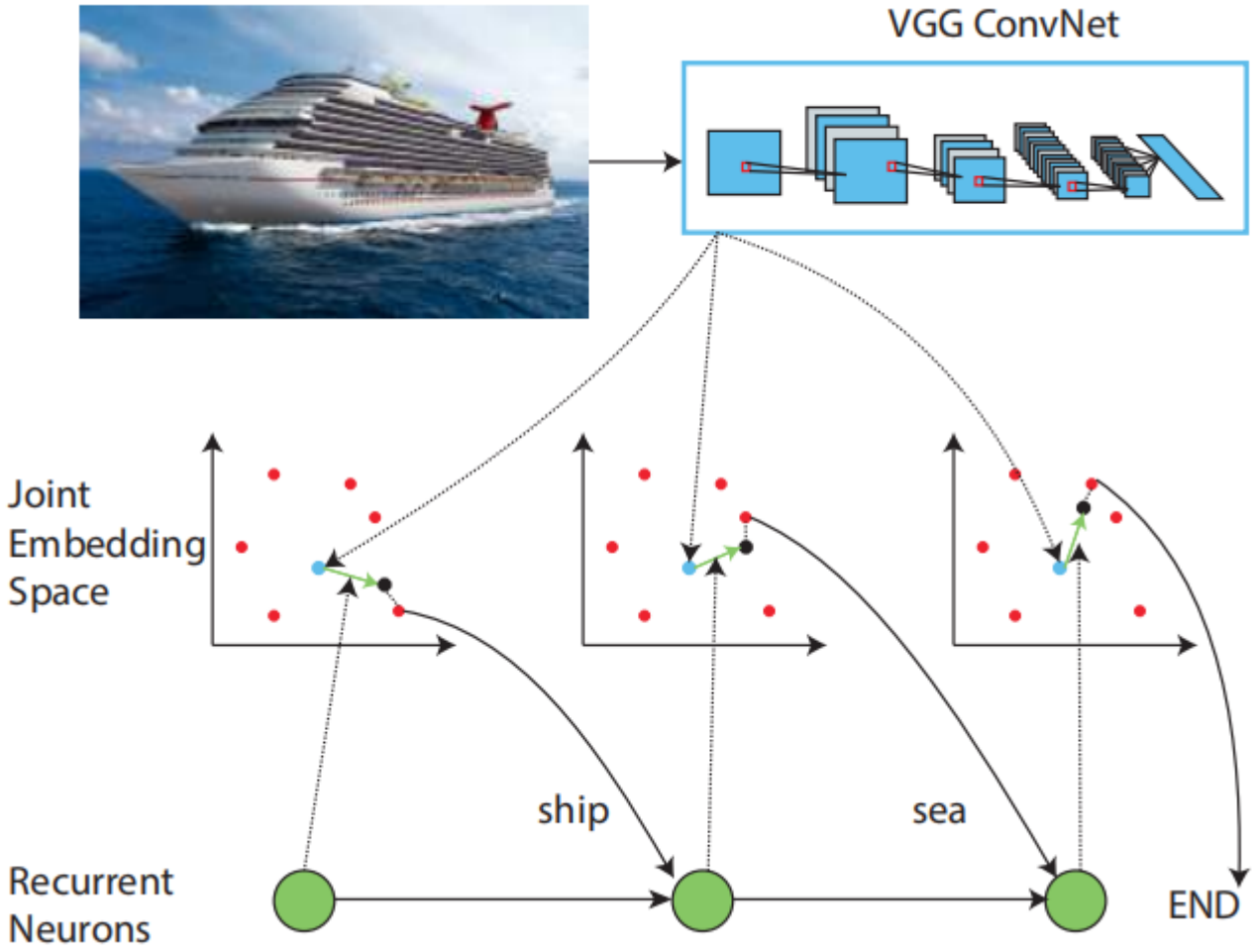
$$o(t) = h_o(r(t-1), w_k(t)),\ r(t) = h_r(r(t-1), w_k(t))$$

$$x_t = h(U_o^x o(t) + U_I^x I),$$

$$s(t) = U_l^T x_t.$$



The red and blue dots are the label and image embeddings, respectively, and the black dots are the sum of the image and recurrent neuron output embeddings.



Inference

$$
\begin{aligned}
l_1, \cdots, l_k &= \arg\max_{l_1, \cdots, l_k} P(l_1, \cdots, l_k | I) \\
&= \arg\max_{l_1, \cdots, l_k} P(l_1 | I) \times P(l_2 | I, l_1). \\
&\cdots P(l_k | I, l_1, \cdots, l_{k-1})
\end{aligned}
$$

Beam Search:

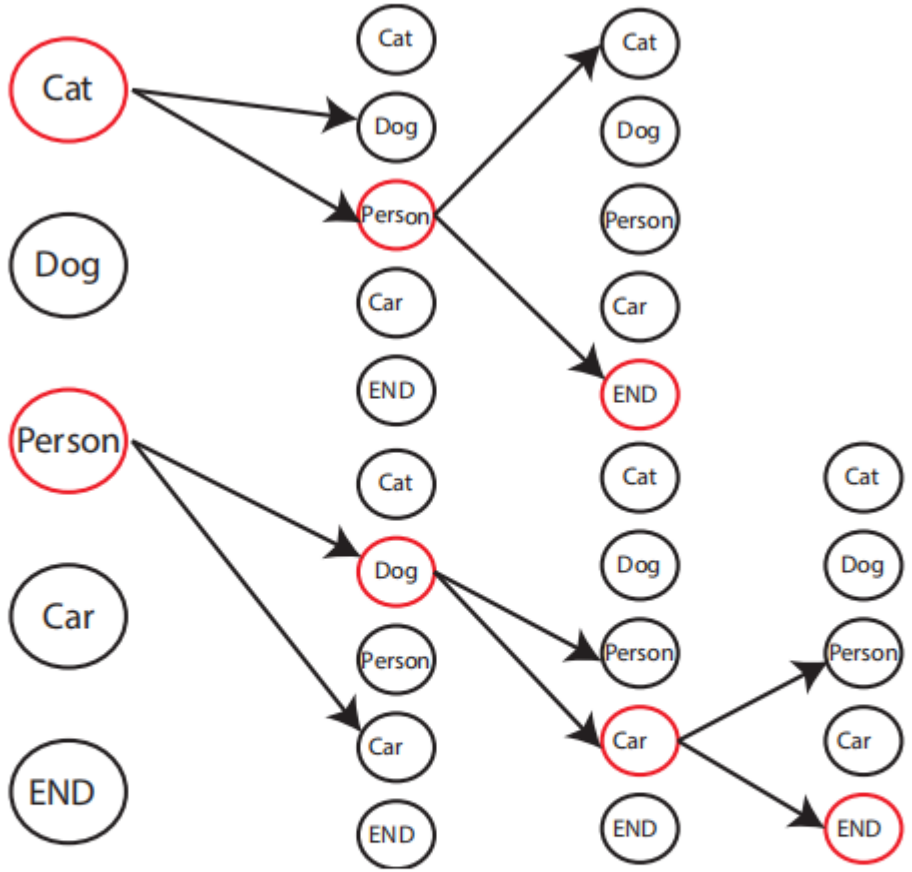


Figure 5. An example of the beam search algorithm with beam size $N = 2$. The beam search algorithm finds the best N paths with the highest probability, by keeping a set of intermediate paths at each time step and iteratively adding labels these intermediate paths.

Expriments

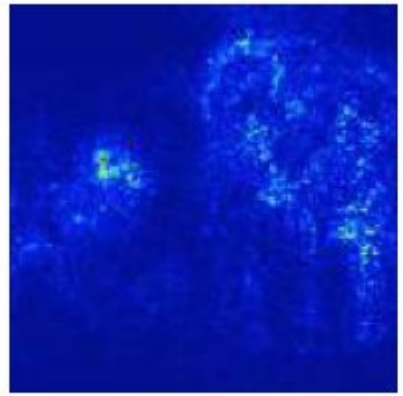
Method	C-P	P-R	C-F1	O-P	O-R	O-F1	MAP@10
Metric Learning [19]	-	-	-	-	-	21.3	-
Multi-edge graph [23]	-	-	-	35.0	37.0	36.0	-
KNN [2]	32.6	19.3	24.3	42.9	53.4	47.6	-
Softmax	31.7	31.2	31.4	47.8	59.5	53.0	-
WARP [9]	31.7	**35.6**	33.5	48.6	60.5	53.9	-
Joint Embedding [38]	-	-	-	-	-	-	40.3
CNN-RNN	**40.5**	30.4	**34.7**	**49.9**	**61.7**	**55.2**	**56.1**

Table 1. Comparisons on NUS-WIDE Dataset on 81 concepts for $k = 3$.

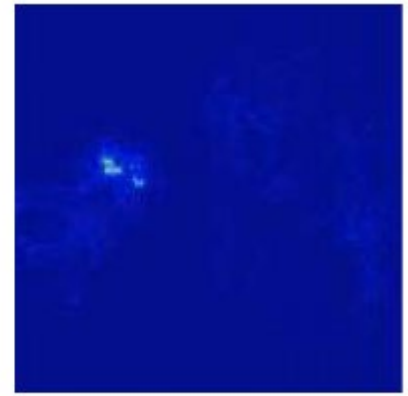
Method	C-P	P-R	C-F1	O-P	O-R	O-F1	MAP@10
Softmax	14.2	**18.6**	16.1	17.1	28.8	21.5	24.3
DLSR [22]	-	-	-	20.0	25.0	22.4	-
WARP	14.5	15.9	15.2	18.3	30.8	22.9	24.8
CNN-RNN	**19.2**	15.3	**17.1**	**18.5**	**31.2**	**23.3**	**26.6**

Table 2. Comparisons on NUS-WIDE Dataset on 1000 tags for $k = 10$.

Original Image



Initial Attention



Attention after first word

Method	C-P	P-R	C-F1	O-P	O-R	O-F1	MAP@10
Softmax	59.0	57.0	58.0	60.2	62.1	61.1	47.4
WARP	59.3	52.5	55.7	59.8	61.4	60.7	49.2
Binary cross-entropy	59.3	**58.6**	58.9	61.7	65.0	63.3	-
No RNN	65.3	54.5	59.3	68.5	61.3	65.7	57.2
CNN-RNN	**66.0**	55.6	**60.4**	**69.2**	**66.4**	**67.8**	**61.2**

Table 3. Comparisons on MS-COCO Dataset for $k = 3$.

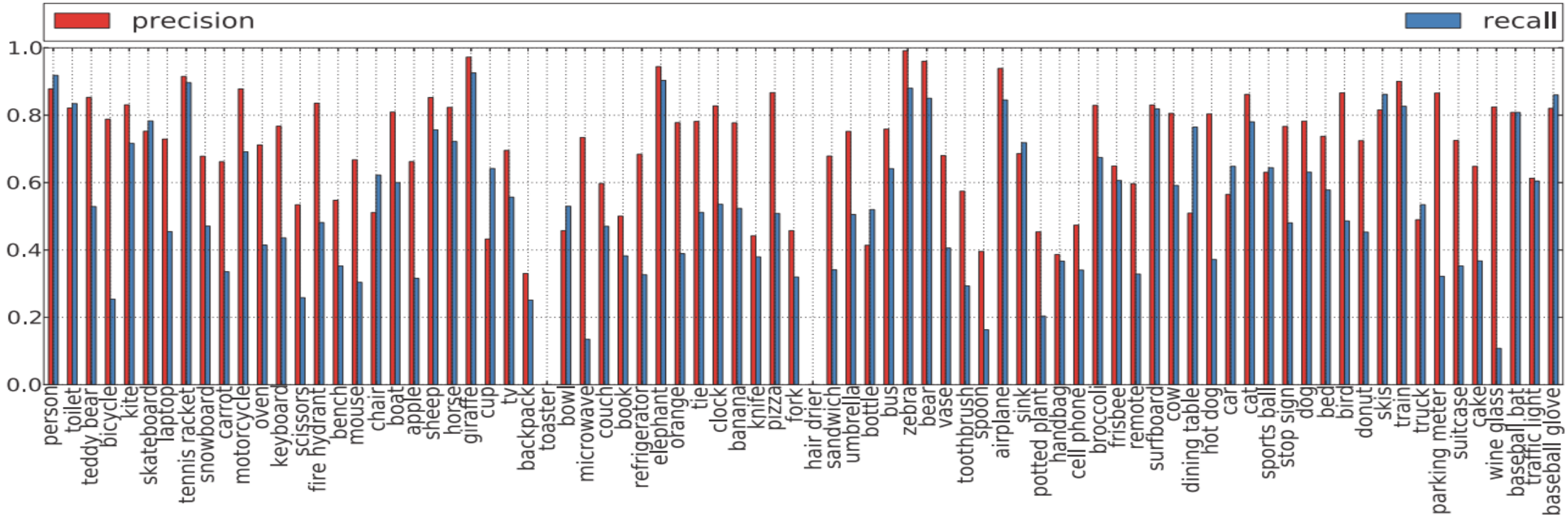


Figure 7. The per-class precision and recall of the RNN model on MS-COCO dataset.

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
INRIA [14]	77.2	69.3	56.2	66.6	45.5	68.1	83.4	53.6	58.3	51.1	62.2	45.2	78.4	69.7	86.1	52.4	54.4	54.3	75.8	62.1	63.5
CNN-SVM [27]	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.4	71.8	73.9
I-FT [36]	91.4	84.7	87.5	81.8	40.2	73.0	86.4	84.8	51.8	63.9	67.9	82.7	84.0	76.9	90.4	51.5	79.9	54.1	89.5	65.8	74.4
HCP-1000C [36]	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
CNN-RNN	**96.7**	83.1	**94.2**	**92.8**	**61.2**	**82.1**	89.1	**94.2**	**64.2**	**83.6**	70.0	**92.4**	**91.7**	84.2	**93.7**	59.8	**93.2**	**75.3**	**99.7**	**78.6**	**84.0**

Table 4. Classification results (AP in %) comparison on PASCAL VOC 2007 dataset.