VALSE 2019

April 11 – 14, 2019

Recognition from Web Data: A Progressive Filtering Approach

Jufeng Yang[1], Xiaoxiao Sun[1], Yu-Kun Lai[2], Liang Zheng[3], Ming-Ming Cheng[1]
[1]Nankai University, [2]Cardiff University , [3]Australian National University
Email: sunxiaoxiaozrt@163.com
Website: cv.nankai.edu.cn

This paper has been accepted by TIP 2018

Introduction

Leveraging the abundant number of web data is a promising strategy in addressing the problem of data lacking when training convolutional neural networks (CNNs). However, web images often contain incorrect tags, which may compromise the learned CNN model.

Fig. 1. Examples of the Food-101 dataset (**left**) and noisy web data (**right**) from the classes *ice cream* and *frozen yogurt*. On the left are images from the public dataset, and the right shows the images collected from the web by searching with the category names, listed in the decreasing order of reliability. We observe that images in the two middle columns of the web images are ambiguous *w.r.t.* their categories, and those in the rightmost column are outliers.

Our contributions:

- We propose an iterative filtering method that progressively learns from web data to improve the performance of CNN model.
- Instead of hard label assignment, images are assigned with multiple labels, which increases the recall of web training data selection.
- We have collected four web image datasets in correspondence to four public classification tasks. Extensive experiments demonstrate that our method yields competitive recognition accuracy against the state-of-the-art approaches.

Method

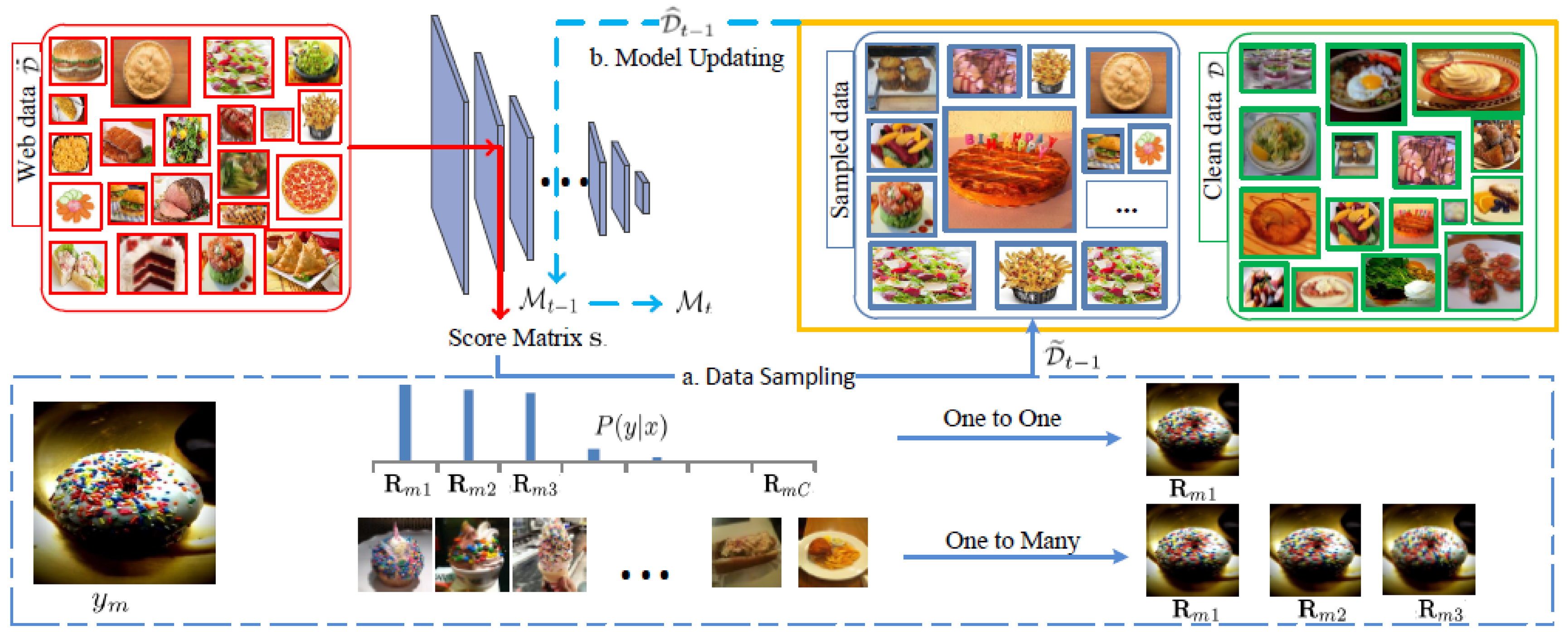


Fig. 2. Pipeline of the proposed progressive learning method, which includes two major steps, namely data sampling and model updating. For data sampling, we obtain $P(y|x)$, score matrix **S** and label matrix **R** of web data based on model $\mathcal{M}_{t-1}$, and use such information to obtain the dataset $\widetilde{\mathcal{D}}_{t-1}$ through a sampling scheme. Here, one to one and one to many are two types of label sampling strategies. For model updating, model $\mathcal{M}_t$ is initialized with the parameters of $\mathcal{M}_{t-1}$ and updated on the combined dataset $\widetilde{\mathcal{D}}_{t-1}$ in the *t*-th iteration.

Algorithm 1 Progressive Learning from Web Data

Input:

Clean dataset: $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$;

The noisy web dataset: $\widehat{\mathcal{D}} = \{(x_m, y_m)\}_{m=1}^{M}$;

The initialized network model $\mathcal{M}_{pre} : f(x; \boldsymbol{\theta}_{pre}) \in \mathbb{R}^C$.

1: Fine-tune a CNN model $\mathcal{M}_0$ based on $\mathcal{M}_{pre}$ using $\mathcal{D}$;
2: Calculate **S** and **R** for $\widehat{\mathcal{D}}$ using model $\mathcal{M}_0$ with Eq. 2;
3: Update $\widehat{\mathcal{D}}$ to obtain $\widetilde{\mathcal{D}}_0$ with Eq. 5;
4: $t \longleftarrow 0$;
5: **repeat**
6: $t \longleftarrow t+1$;
7: **repeat**
8: Update parameters $\boldsymbol{\theta}_t^*$ with Eq. 4 on each mini-batch $\widehat{\mathcal{D}}_{t-1}^b$, $\widehat{\mathcal{D}} = \mathcal{D} \cup \widetilde{\mathcal{D}}_{t-1}$;
9: **until** loss function $L(x, y)$ has converged.
10: Obtain model $\mathcal{M}_t$;
11: Calculate **S** and **R** for $\widehat{\mathcal{D}}$ using $\mathcal{M}_t$ based on Eq. 2;
12: Update the dataset $\widehat{\mathcal{D}}$ to obtain $\widetilde{\mathcal{D}}_t$ with Eq. 5;
13: **until** $\widetilde{\mathcal{D}}_t$ tends to be stable or the performance of $\mathcal{M}_t$ does not improve.

Output:

The trained model: $\mathcal{M}_t : f(x; \boldsymbol{\theta}_t) \in \mathbb{R}^C$.

A. Progressive Learning from Noisy Web Labels

Web data selection for CNN training: We use a softmax function to generate the probability for each web image x_m :

$$p(l_i|x_m) = \frac{e^{\boldsymbol{\theta}_i^\top x_m}}{\sum_j e^{\boldsymbol{\theta}_j^\top x_m}}. \quad (2)$$

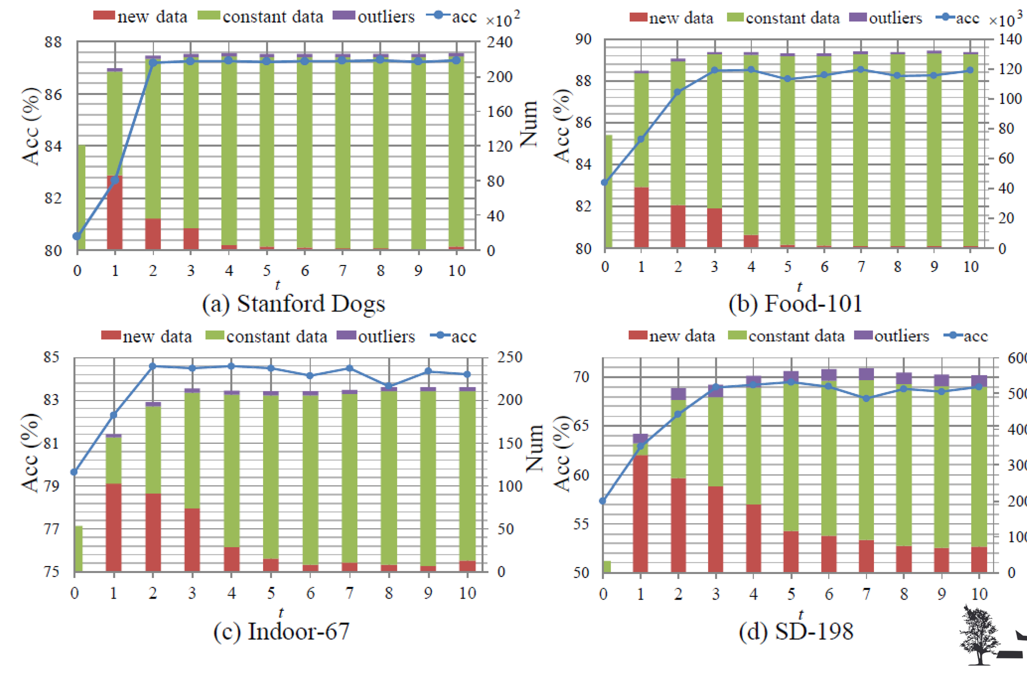
B. One-to-Many Correction for Noisy Labels

One-to-many label assignment: We replace the one-to-one label assignment method with one-to-many scheme:

$$\widehat{y}_m = \begin{cases} \mathbf{R}_{m1}, & \text{case 1} \\ \{\mathbf{R}_{mi}\}_{i=1}^{k}, & \text{case 2} \\ \emptyset, & \text{otherwise} \end{cases} \quad (5)$$

Experiment

Classification accuracies (Acc) of different iterations.



Ablation study.

K	Num	Percentage	Num+	Acc (%)
0	6881	56.14	0	79.70
1	7020	57.28	0	80.23
2	8674	70.78	1654	**82.47**
3	9491	77.45	3288	82.39
4	9930	81.03	4650	82.43
5	10238	83.54	5837	81.98

Performance on different scales of clean training set on STANFORD DOGS.

i	1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80	90
$\mathcal{D}$+ft	16.35	30.76	39.78	50.49	55.72	59.91	62.24	63.52	66.30	66.85	73.02	74.75	77.15	77.98	78.21	78.79	78.87	79.47
$\widehat{\mathcal{D}}$+ft	**74.58**	75.07	75.59	76.46	76.72	77.37	77.62	76.67	77.43	75.35	78.11	78.89	78.42	78.94	80.10	79.75	80.01	80.71
Ours	73.17	**76.64**	**77.21**	**79.99**	**80.44**	**80.78**	**80.38**	**80.81**	**80.74**	**81.26**	**82.38**	**82.53**	**83.01**	**83.56**	**84.12**	**85.33**	**85.79**	**86.34**

Accuracies (%) on four datasets with different methods.

#	Type	Method	Model	Test Accuracy				
				SD-198	Food-101	Dogs-120	Indoor-67	Dogs-120 (+L-Dog)
1	Baseline	$\mathcal{D}_{clean}$+ft	Alexnet	50.85	65.93	63.57	65.52	63.57
2		$\mathcal{D}_{mix}$ +ft		42.71	69.71	65.63	69.63	64.84
3		$\mathcal{D}_{filter}$+ft		50.92	69.89	67.95	63.58	66.16
4	Previous work	Bottom-up [53]		44.26	70.29	72.17	64.18	71.59
6		Pseudo-label [54]		44.02	69.36	70.32	63.88	71.01
7		Weakly [55]		45.44	71.10	73.00	64.85	73.64
8	Ours	Baseline+A		52.76	73.81	72.30	69.63	71.42
9		Baseline+B		51.06	70.64	69.43	65.47	69.35
10		Baseline+A+B		53.46	73.78	73.99	70.30	73.20
11		Baseline+A+B+∗		**54.37**	**75.65**	**75.52**	**71.12**	**74.93**
12	Baseline	$\mathcal{D}_{clean}$+ft	Caffenet	51.34	66.61	63.19	65.22	63.19
13		$\mathcal{D}_{mix}$ +ft		44.74	69.25	66.08	67.99	65.34
14		$\mathcal{D}_{filter}$+ft		51.43	68.48	69.56	63.51	65.90
15	Previous work	Boosting [56]		45.99	72.53	73.49	65.60	73.28
16		PGM [17]		46.38	73.14	72.63	65.30	71.83
17		WSL [6]		46.99	73.21	73.52	65.60	73.79
18	Ours	Baseline+A		51.97	73.57	73.05	68.21	72.64
19		Baseline+B		51.55	71.13	72.34	65.45	71.22
20		Baseline+A+B		53.01	75.24	73.68	69.03	73.80
21		Baseline+A+B∗		**53.95**	**76.92**	**75.74**	**71.19**	**75.63**
22	Baseline	$\mathcal{D}_{clean}$+ft	VGGNet	55.19	74.32	78.29	71.79	78.68
23		$\mathcal{D}_{mix}$ +ft		51.58	76.98	81.03	72.01	77.54
24		$\mathcal{D}_{filter}$+ft		53.31	78.24	79.70	72.16	79.57
25	Previous work	Harnessing [12]		54.50	79.02	78.45	70.00	78.31
26	Ours	Baseline+A		56.23	79.59	83.12	72.46	81.95
27		Baseline+B		55.47	78.71	82.47	72.24	80.66
28		Baseline+A+B		57.44	79.93	83.72	73.51	82.51
29		Baseline+A+B+∗		**59.66**	**81.32**	**84.36**	**75.97**	**83.75**
30	Baseline	$\mathcal{D}_{clean}$+ft	Resnet50	57.35	83.14	80.51	79.63	80.51
31		$\mathcal{D}_{mix}$ +ft		54.22	85.21	81.43	82.31	82.07
32		$\mathcal{D}_{filter}$+ft		63.49	86.10	82.62	81.34	83.61
33	Previous work	Goldfince [11]		65.74	86.75	85.90	83.43	85.48
34	Ours	Baseline+A		65.67	88.58	84.57	82.54	84.57
35		Baseline+B		64.19	86.47	83.26	82.24	83.93
36		Baseline+A+B		67.25	88.96	85.93	83.58	85.69
37		Baseline+A+B+∗		**70.56**	**89.77**	**87.36**	**84.78**	**86.94**

Comparison with state-of-the-art methods on four datasets.

SD-198		Food-101		Stanford Dogs		MIT Indoor	
Method	Acc (%)	Method	Acc (%)	Method	Acc (%)	Method	Acc (%)
Caffe [43]	42.31	Random Forest [45]	50.76	NAC [57]	68.61	IFV+DMS [58]	66.87
VGG [43]	37.91	SNN [59]	69.90	FoF-Weakly [60]	71.40	FB/REF [12]	61.60
Caffe+ft [43]	46.69	DCNN [45]	56.40	PDFS [61]	71.96	CL-45C [62]	68.80
VGG+ft [43]	50.27	CNNFM [63]	58.49	FB/REF [12]	73.10	MLVED [64]	69.69
NPT [65]	52.19	DCNN+ft [45]	68.44	FOAF+ft [66]	74.49	Hybrid-CNN [67]	70.80
CSDR [68]	56.47	PTFT [69]	70.41	MagNet [70]	75.10	CNN+G [64]	70.46
Ours (Resnet50)	**70.56**	Im2Calories [71]	79.00	RED-OSSVR(vs) [72]	79.50	S-NN [59]	72.20
		ResNet50+ft	84.31	Weakly-S [73]	80.43	SFV [74]	72.86
		ResNet110+ft	84.88	Inception-v3 [11]	80.60	MPP+DSFL [75]	80.78
		Inception-v3 [76]	88.28	Goldfince [11]	85.90	Double fully hybrid [77]	80.97
		Ours (Resnet50)	**89.77**	Ours (Resnet50)	**87.36**	Ours (Resnet50)	**84.78**