Scalable Logo Detection by Self Co-Learning

Hang Su[a,*], Shaogang Gong[a], Xiatian Zhu[b]

[a] *Queen Mary University of London, London E1 4NS, UK*
[b] *Vision Semantics Limited, London E1 4NS, UK*

Abstract

Existing logo detection methods usually consider a small number of logo classes, limited images per class and assume fine-gained object bounding box annotations. This limits their scalability to real-world dynamic applications. In this work, we tackle these challenges by exploring a web data learning principle without the need for exhaustive manual labelling. Specifically, we propose a novel incremental learning approach, called Scalable Logo Self-co-Learning (SL^2), capable of automatically self-discovering informative training images from noisy web data for progressively improving model capability in a cross-model co-learning manner. Moreover, we introduce a very large (2,190,757 images of 194 logo classes) logo dataset "WebLogo-2M" by designing an automatic data collection and processing method. Extensive comparative evaluations demonstrate the superiority of SL^2 over the state-of-the-art strongly and weakly supervised detection models and contemporary web data learning approaches.

Keywords: Object Detection; Logo Recognition; Logo Dataset; Web Data Mining; Self-Learning; Co-Learning.

1. Introduction

Automated logo detection from unconstrained "in-the-wild" images benefits a wide range of applications, document image logo retrieval [1] and vehicle logo recognition in intelligent transportation [2]. This is inherently a challenging task due to the pres-

*Corresponding author; Phone, +44 7576498032; London E1 4NS, UK
Email addresses: `hang.su@qmul.ac.uk` (Hang Su), `s.gong@qmul.ac.uk` (Shaogang Gong), `eddy.zhuxt@gmail.com` (Xiatian Zhu)

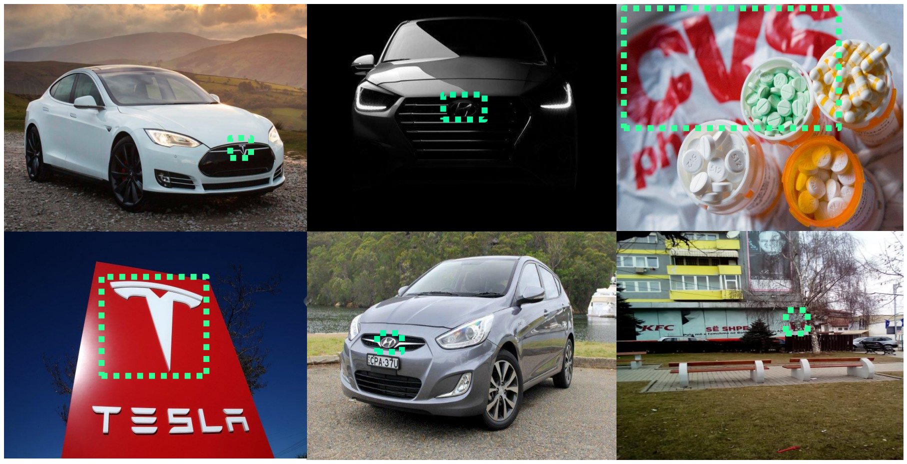

Figure 1: Logo detection challenges: significant variations in scale, illumination, background, and occlusion.

ence of many logos in diverse context with uncontrolled illumination, varying scales, occlusion, low-resolution, and background clutter (Fig. 1).

Existing logo detection methods typically consider a small number of logo classes with the need for large scale labelled training data at the object instance level [3]. Whilst this controlled setting allows for a straightforward adoption of the state-of-the-art object detection models such as Faster R-CNN [4] and YOLO [5], it is non-scalable to real-world logo detection applications when a much larger number of logo classes are targeted. This is due to two reasons: (1) Extremely high cost for constructing large scale dataset with exhaustive logo instance bounding box labelling [6]; (2) Lacking the incremental model learning ability to progressively update and expand the model to increasingly more training data without fine-grained labelling. Existing models are mostly one-pass trained with limited generalisation to new classes.

Table 1: Statistics and characteristics of existing logo detection benchmarking datasets.

Dataset	Logo Classes	Images	Supervision	Noisy	Construction	Scalability	Availability
TopLogo-10 [7]	10	700	Object-Level	✗	Manually	Weak	✓
TennisLogo-20 [8]	20	2,000	Object-Level	✗	Manually	Weak	✗
FlickrLogos-27 [9]	27	810	Object-Level	✗	Manually	Weak	✓
FlickrLogos-32 [10]	32	2,240	Object-Level	✗	Manually	Weak	✓
Logo32-270 [11]	32	8,640	Object-Level	✗	Manually	Weak	✗
BelgaLogos [12]	37	1321	Object-Level	✗	Manually	Weak	✓
LOGO-NET [13]	160	73,414	Object-Level	✗	Manually	Weak	✗
Logo-In-The-Wild [14]	1196	9,393	Object-Level	✗	Manually	Weak	✓
SportsLogo [8]	20	1,978	Object-Level	✗	Manually	Weak	✓
MICC-Logos [15]	13	720	Object-Level	✗	Manually	Weak	✗
WebLogo-2M (Ours)	**194**	**2,190,757**	Image-Level	✓	Automatically	Strong	✓

In this work, we consider the problem of scalable logo detection learning in a very large collection of unconstrained images without exhaustive fine-grained instance level labelling. Given that the existing datasets mostly have small numbers of logo classes, one possible strategy is to learn from a small set of labelled training classes and then adopt the model to other novel (test) logo classes, that is, Zero-Shot Learning (ZSL) [16]. This class-to-class model transfer and generalisation in ZSL is achieved by knowledge sharing through an intermediate semantic representation for all classes, such as mid-level attributes [16] or a class name embedding space [17]. However, they are limited as many logos do not share attributes or other forms of semantic representations due to their unique A lack of large scale logo datasets (Table 1), in both class size and per-class image number severely limits the scalability of current logo detection models. This study explores a web data learning principle for both large scale dataset construction and incremental logo detection model learning without exhaustive manual annotation on increasing logo data. The aim is to scale up the limited logo detection capacity to large dynamic real-world applications by exploiting the rich multimedia data from the Internet. We call this setting *scalable logo detection*.

The **contributions** of this work are three-fold: **(1)** We investigate the scalable logo detection problem, characterised by modelling a large quantity of logo classes *without* exhaustive bounding box annotation. This is different from the existing methods typically considering only a small number of logo classes with the need for manual labelling. This scalability problem is under-studied in the literature. **(2)** We propose a novel incremental learning approach to scalable logo detection by exploiting multi-class detection with context enhancement. We call this method *Scalable Logo Self-co-Learning* (SL^2), since it automatically discovers potential positive logo images from noisy web data to progressively improve the model discrimination and generalisation capability in a *self-learning and co-learning* manner. **(3)** We introduce a large logo dataset including 2,190,757 images from 194 logo classes, called *WebLogo-2M*, created by *automatically* sampling web logo images from the Twitter website. Importantly, our construction method allows to further expand the dataset easily with new logo classes and images, therefore offering a favourable solution for Extensive experiments demonstrate the superiority of SL^2 over the state-of-the-art strongly (Faster R-CNN [4], SSD

[18], RetinaNet [19], YOLOv2 [5], and YOLOv3 [20]) and weakly (WSL [21], PCL [22]) supervised detection models, and webly learning methods (WLOD [23]) on the WebLogo-2M dataset[1].

The preliminary version of this has been reported in [24]. Compared with the earlier study, there are several key differences introduced: (i) This study presents a more advanced method by introducing a joint co-training and self-learning concept into the scalable logo detection model formulation. This enables mining the complementary advantages of two different detection models, making self-learning significantly more effective. (ii) We conduct more comprehensive evaluations and analysis on incremental model learning in this study for giving more insights. (iii) We further expand the large WebLogo-2M dataset by additional data collection and manual labelling.

2. Related Works

Logo Detection Early logo detection methods are established on hand-crafted visual features (e.g. SIFT [25] and HOG [3]) and conventional classification models (e.g. BoW [26]). These methods were only evaluated by small logo datasets with a limited number of logo images and classes. Recently, Convolutional Neural Networks (CNN) have emerged as stronger solutions [27]. A few deep logo detection methods [7, 28, 29] have been recently proposed by exploiting the state-of-the-art object detection models such as Faster R-CNN [4]. This leads to a need for a large number of labelled training data. To this end, a couple of works leverage many synthetic logo imagery with the bounding boxes obtained at zero annotation cost [28, 7]. To better generalise logo detection, the notions of universal logo detection [29, 14] and open set logo retrieval [14] have been formulated respectively. Meanwhile, this also inspires large data construction [13]. However, all these existing models are not scalable to real world deployment due to two stringent requirements: (1) Accurately labelled training data per logo class; (2) Strong object-level bounding box annotations. This is because, both requirements give rise to time-consuming training data collection and annotation, which is

[1] The WebLogo-2M benchmark is released publicly at: `https://weblogo2m.github.io/`.

not scalable to a very large number of logo classes given limited human labelling budget. In contrast, our method eliminates both needs by enabling model learning from image-level weakly annotated and noisy web images. As such, we enable automated introduction of any quantity of new logos for both dataset construction/expansion and model update without exhaustive manual labelling.

Logo Datasets A number of logo detection datasets exist in the literature (Table 1). All existing datasets are constructed *manually* and typically small in both sample and category thus insufficient for deep learning. Recently, Hoi et al. [13] attempt to create a large scale logo dataset LOGO-NET. However, it is still not publicly accessible. To address this scalability problem, we propose to collect logo images *automatically* from the social media. This brings about two unique benefits: (1) Weak image level labels can be obtained for free; (2) We can easily upgrade the dataset by expanding the logo category set and collecting new logo images without human labelling therefore scalable to any quantity of logo images and categories. To our knowledge, this is the first attempt to construct a large scale logo dataset by exploiting inherently noisy web data.

Model Self-Learning Self-training is a special type of incremental learning where the new training data are labelled by the model itself – predicting logo positions and class labels in weakly labelled or unlabelled images before converting the most confident predictions into the training data [30]. A similar approach to our model is the detection model by Rosenberg et al. [31]. This model also explores the self-training mechanism. However, this method needs a number of per-class strongly and accurately labelled training data to initialise the detection model. Also, it assumes unlabelled images drawn from the target categories. Such assumptions severely limit the model usability and scalability when only noisy web training data are available. **Model Co-Learning** Model co-learning is a generic learning strategy originally designed for semi-supervised learning, based on two sufficient and conditionally independent feature representations with a single model algorithm [32]. Later on, co-learning was further developed into the variants of using different model parameter settings [33] or models [34] on the same feature representation. Recently, this strategy is also applied for hyperspectral data classification by co-training of spectral and spatial information

[35], and multi-source domain adaptation by co-regression [36]. Overall, the key is that both models in co-learning need be independently effective and complementary to each other. Beyond these, we further extend the co-learning concept from semi-supervised learning to web data learning for scalable logo detection. In particular, we unite co-learning and self-learning in a single detection deep learning framework with the capability of incrementally improving logo detection models. To our knowledge, this is the first attempt of exploiting such a *self-co-learning* approach in the logo detection literature.

3. WebLogo-2M Logo Detection Dataset

We present a scalable method to automatically construct a large logo dataset, called *WebLogo-2M*, including 2,190,757 web images from 194 classes (Table 2).

Table 2: WebLogo-2M statistics. Numbers in parentheses: the minimum/median/maximum per class.

Logos	Raw Images	Filtered Images	Noise Rate (%)
194	4,941,317	2,190,757	Varying
-	-	(6/2583/179,789)	(25.0/90.2/99.8)

3.1. Logo Image Collection and Filtering

Logo Selection A total of 194 logo classes from 13 different categories are selected in the WebLogo-2M dataset (Fig. 4). They are popular logos and brands in our daily life, including 32 logo classes of FlickrLogo-32 [10] and 10 logo classes of TopLogo-10 [7]. Specifically, the logo class selection was guided by an extensive review of social media reports regarding to brand popularity [2][3][4] and market-value[5][6].

Image Source Selection We selected the social media website Twitter as the data source of WebLogo-2M. Twitter offers well structured multi-media data stream sources

[2] http://www.ranker.com/crowdranked-list/ranking-the-best-logos-in-the-world

[3] http://zankrank.com/Ranqings/?currentRanqing=logos

[4] http://uk.complex.com/style/2013/03/the-50-most-iconic-brand-logos-of-all-time

[5] http://www.forbes.com/powerful-brands/list/#tab:rank

[6] http://brandirectory.com/league_tables/table/apparel-50-2016

and more critically, unlimited data access permission therefore facilitating the collection of large scale logo images. We also attempted with Google and Bing search engines, and three other social media websites (Facebook, Instagram, and Flickr). However, all of them are more restricted in data access and limiting incremental big data collection, for example, Instagram allows only 500 times of image downloading per hour through the official web API. The Amazon website provides a rich logo imagery source but limited to constrained product images with clean background.

Image Collection We collected 4,941,317 web logo images. Specifically, through the Twitter API, one can automatically retrieve images from tweets by matching query keywords. In our case, we query the logo names so that images in tweets containing the query words can be extracted. The collected images are then labelled with the corresponding logo name at the image level, i.e. *weakly labelled*.

Logo Image Filtering We obtained a total of 2,190,757 images after conducting a two-steps auto-filtering: (1) *Noise Removal*: We removed images of small width and/or height (e.g. less than 100 pixels), statistically we observed that such images are mostly without any logo objects (noise). (2) *Duplicate Removal*: We identified and discarded duplicates. Specifically, given a reference image, we removed those with identical width and height. This image spacial size based scheme is not only computationally cheaper than the appearance matching alternative [37], but also effective. For example, we manually examined the de-duplicating process on 50 randomly selected reference images and found that over 90% of the images are true duplicates.

3.2. Properties of WebLogo-2M

Compared to existing logo datasets like FlickrLogos-32 [10], LOGO-NET [13] and TopLogo-10 [7], this web logo image dataset presents three *distinct* properties inherent to large scale data exploration for learning scalable logo models:

(I) Weak Annotation All WebLogo-2M images are weakly labelled at the image level. Since the labels are obtained automatically, it is much more scalable than those with the need for manual annotation of logo bounding boxes, particularly when logo images and classes are at large scales.

(II) Noisy (False Positives) Web images are inherently noisy with most presenting no logo classes, therefore exhibiting plenty of false positive samples. For estimating the noise degree, we sampled randomly and examined manually up to 1,000 web images per class[7]. As shown in Fig. 2, the true logo image ratio varies significantly over classes, e.g. 75% for "Rittersport" vs. 0.2% for "3M". On average, only 21.26% of the examined imagery are true positives. Such noisy images pose significant challenges to model learning, even though there are plenty of training data.

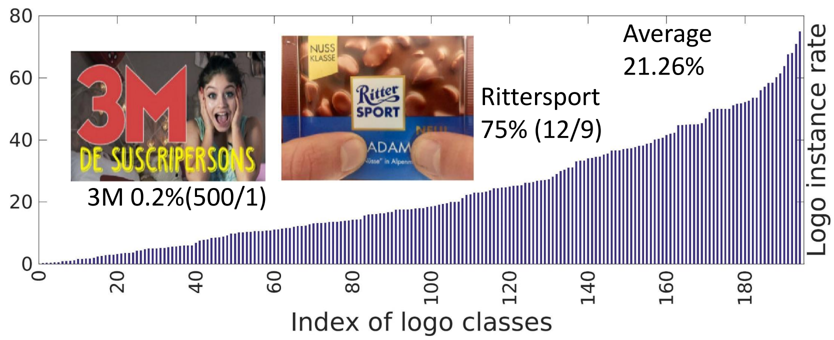

Figure 2: True logo image ratios (%). This was estimated from up to 1,000 random images per class.

(III) Class Imbalance The WebLogo-2M dataset presents a natural logo object occurrence imbalance in public scenes. Specifically, logo images collected from web streams exhibit a power-law distribution (Fig. 3). This property is often artificially eliminated in most existing logo datasets by careful manual filtering, which not only requires extra labelling effort but also renders the model learning challenges *unrealistic*. We preserve the inherent class imbalance nature for achieving fully automated dataset construction and retaining realistic model learning challenges. This requires minimising model learning bias towards densely-sampled classes [38].

Further Remarks Since the proposed dataset construction method is completely automated, new logo classes can be easily added without human labelling. This permits scalability for facilitating dataset expansion, in contrast to existing methods of ImageNet [6], PASCAL VOC [39], MSCOCO [40] that require exhaustive human la-

[7] For sparse logo classes with $<$1,000 web images, we examined the whole.

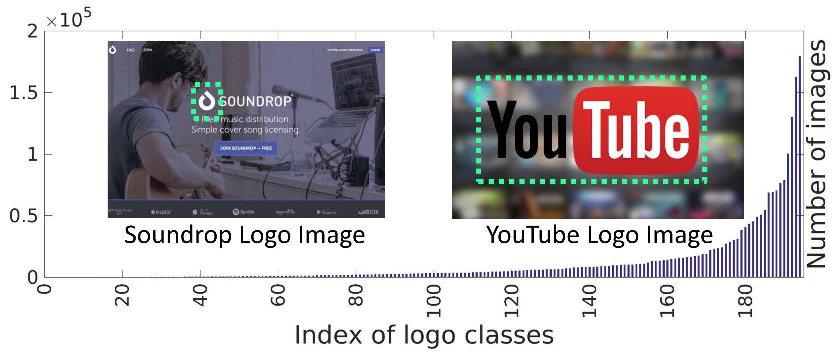

Figure 3: Imbalanced logo image class distribution, ranging from 6 images ("Soundrop") to 179,789 images ("Youtube"), with the imbalance ratio as severe as 1:29,965.

belling. This automation is particularly more important for object detection datasets with expensive needs for labelling bounding boxes, beyond cheaper image-level class label annotation [41]. While being more scalable, WebLogo-2M poses more realistic challenges to model learning due to weaker label information, noisy image data, unknown scene context, and significant class imbalance.

3.3. Benchmarking Training and Test Data

We define a benchmarking logo detection setting here. In the scalable webly learning context, we deploy the whole WebLogo-2M dataset (2,190,757 images) as the *training* data. For performance evaluation, a set of images with bounding box annotation groundtruth is required. To that end, we construct an independent *test set* of 6,558 logo images with logo bounding box labels by (1) assembling 2,870 labelled images from the FlickrLogo-32 [10] and TopLogo [7] datasets and (2) manually labelling 3,688 images independently collected from the Twitter website. Note that, the test set is only for model performance evaluation, independent of WebLogo-2M auto-construction.

4. Training A Multi-Class Logo Detector

We aim to automatically train a multi-class logo detection model from noisy and weakly labelled web images. Different from existing methods building a detector in a one-pass "batch" learning procedure, we propose to incrementally enhance the model

Figure 4: A glimpse of the WebLogo-2M dataset. **(a)** Example webly (Twitter) logo images randomly selected from the class "Adidas" with logo instances manually labelled by green dashed bounding boxes only for facilitating viewing. Most images contain no "Adidas" object, i.e. false positives. This suggests a high noise degree in such webly collected data without exhaustive filtering and selection. **(b)** Clean images of 194 logo classes automatically collected from the Google Image Search, used in synthetic training images generation and context enhancement. **(c)** Examples of true positive web images per logo class, totally 194 images, showing the rich and diverse context in unconstrained images where typical logo objects reside in practice, as compared to those clean logo images in (b).

capability in a joint spirit of self-learning [30] and co-learning [32]. This is due to the *unavailability* of sufficient accurate fine-grained training data. In particular, the model must self-select reliable images from the noisy WebLogo-2M to progressively develop and refine itself. This is a catch-22 problem: The lack of sufficient good-quality training data leads to a suboptimal model that is error-prone during inference. This may cause *model drift* – the errors in model prediction will be propagated and cumulated through the iterations therefore have the potential to corrupt the model knowledge structure. Also, the inherent class imbalance may make model learning biased towards only a few number of majority classes whilst neglecting the minority classes. The two problems above are intrinsically interfered. It is non-trivial to solve these challenges without exhaustive fine-grained manual annotations of training data.

Formulation Rationale In this work, we present a scalable logo detection solution capable of addressing the aforementioned two issues in a self-co-learning manner. The intuition is that, web knowledge provides ambiguous and useful image level logo annotations, self-learning offers a scalable learning mechanism to explore such information and co-learning allows for mining the complementary advantages of different models in order to further improve the effectiveness of self-learning. Note that self-mining of training data may introduce label errors which can further propagate and expand through training. To better leverage co-learning, it is favoured that two learners differ significantly with certain conditional independence and respective specificity. As such, they can achieve jointly high complementary effects to mutually benefit each other. We call the proposed method *Scalable Logo Self-co-Learning* (SL^2).

Model Design To establish a more effective SL^2 framework, we select strongly-supervised rather than weakly-supervised object detection models for two reasons: (1) Weakly-supervised models [42] are much inferior; (2) The noisy labels may further hamper the efficacy of weakly supervised learning. In our self-co-learning instantiation, we choose the Faster R-CNN [4] and YOLOv2 [5] models based on two considerations: (1) Faster R-CNN and YOLOv2 are formulated by different design principles with good complementary hence suitable for co-learning. (2) We empirically found that the two models perform superiorly for scalable logo detection as compared to arguably

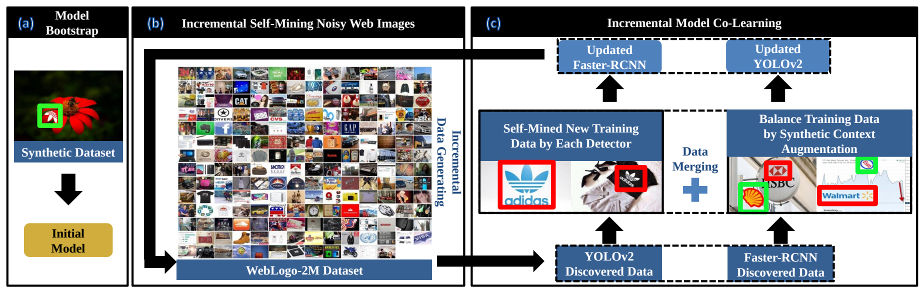

Figure 5: Overview of the Scalable Logo Self-co-Learning (SL^2) method. **(a)** Model initialisation by using synthetic logo training images (Sec. 4.1). **(b)** Incrementally self-mining positive logo images from noisy web data pool (Sec. 4.2). **(c)** Incrementally co-learning the detection models by mined web images and context-enhanced synthetic data (Sec. 4.3). This process is repeated iteratively for progressive training data mining and model update.

stronger alternatives RetinaNet with FPN [19], YOLOv3 [20] and SSD [18] (see Table 3). Note, this model selection is conceptually independent of the SL^2 formulation. A schematic overview of SL^2 is depicted in Fig. 5.

4.1. Model Bootstrap

To start the SL^2 process, we feed logo detection model co-learning with bootstrapping training data. Both Faster R-CNN and YOLOv2 need supervised learning from bounding box annotations to achieve detection discrimination, which however is not available in our webly learning setting.

To address this problem above in our context, we exploit the idea of synthesising fine-grained training logo images for maintaining model learning scalability for accommodating large quantity of logo classes. In particular, this is achieved by generating synthetic training images as in [7]: Overlaying *logo icon images* at random locations of non-logo background images so that bounding box annotations can be *automatically* and *completely* generated. The logo icon images are automatically collected from Google Image Search by querying logo class names (Fig. 4 (b)). The background images can be chosen flexibly, e.g. non-logo images in FlickrLogo-32 [10] and others retrieved by irrelevant query words from search engines. To enhance appearance variations in synthetic logos, colour and geometric transformation can be applied [7].

Training Details We synthesised 1000 training images per class, totally 194,000 images. This is estimated based on the cost-effectiveness of YOLOv2 (Table 7). For learning the Faster R-CNN and YOLOv2 models, we set the learning rate at 0.0001 and the learning iterations at $6,000$. Following [7], we pre-trained the models on ImageNet [6] for model warmup.

4.2. Incremental Self-Mining Noisy Web Images

After logo detectors are bootstrapped, we proceed to improve their detection capability with self-mined positive (likely) logo images from WebLogo-2M. To identify the most compatible training images, we define a selection function using the detection score of up-to-date model:

$$S(\mathcal{M}_t, \boldsymbol{x}, y) = S_{\text{det}}(y|\mathcal{M}_t, \boldsymbol{x}) \in [0, 1] \tag{1}$$

where $\mathcal{M}_t$ denotes the t-th iteration model (Faster R-CNN or YOLOv2), $\boldsymbol{x}$ represents a training image with the label $y \in Y = \{1, 2, \cdots, m\}$, and m represents the logo class number. $S_{\text{det}}(y|\mathcal{M}_t, \boldsymbol{x})$ specifies the maximal detection score of $\boldsymbol{x}$ on a logo class y inferred by the model $\mathcal{M}_t$. For reliable logo image discovery, we consider a high threshold detection confidence (0.9 in our experiments) [43] for mitigating the impact of model detection errors. The proposed training data discovery and model incremental learning process is summarised in Algorithm 1.

Through the same self-mining process, we obtain a separate set of updated training data for Faster R-CNN and YOLOv2, denoted as $\mathcal{T}_t^f$ and $\mathcal{T}_t^y$ respectively. This leverages the unique characteristics of different model formulations, region proposal based Faster R-CNN versus grid regression based YOLOv2. It hence creates a satisfactory condition for cross-model co-learning.

4.3. Incremental Model Co-Learning

Given the two up-to-date training sets $\mathcal{T}_t^f$ and $\mathcal{T}_t^y$, we conduct co-learning for detection models (Fig. 5(c)). Specifically, we incrementally update Faster R-CNN model using the set $\mathcal{T}_t^y$ mined by YOLOv2, and vice versa. As such, the complementary advantages can be propagated incrementally in a cross-model manner.

Algorithm 1 Incremental self-mining noisy web logo images

Input: Current model $\mathcal{M}_{t-1}$, Unexplored logo training data $\mathcal{D}_{t-1}$, Self-discovered logo training data $\mathcal{T}_{t-1}$ ($\mathcal{T}_0 = \emptyset$);

Output: Updated self-discovered training data $\mathcal{T}_t$, Updated unlabelled data pool $\mathcal{D}_t$;

Initialisation: $\mathcal{T}_t = \mathcal{T}_{t-1}, \mathcal{D}_t = \mathcal{D}_{t-1}$;

for image i in $\mathcal{D}_{t-1}$

  Apply $\mathcal{M}_{t-1}$ to get the detection results;

  Evaluate image i as a potential positive logo image;

    if Meeting selection criterion

      $\mathcal{T}_t$ = $\mathcal{T}_t \cup \{i\}$;

      $\mathcal{D}_t$ = $\mathcal{D}_t \setminus \{i\}$;

    end if

end for

Return $\mathcal{T}_t$ and $\mathcal{D}_t$.

Recall that the logo images are imbalanced across classes (Fig. 3). This causes biased learning favoured towards well-sampled classes [38]. To address this problem, we propose an idea of cross-class context enhancement. It aims for both exploring the rich context of WebLogo-2M and addressing the imbalanced class problem.

Specifically, we ensure that at least N_{cls} images will be newly introduced into the training data pool in each self-discovery iteration for each detection model. Suppose N^i_{sf} web images are self-discovered for the logo class i (Alg. 1), we generate N^i_{syn} synthetic images where

$$N^i_{\text{syn}} = \max(0, N_{\text{cls}} - N^i_{\text{sf}}). \tag{2}$$

Therefore, we only perform synthetic context enhancement for those classes with less than N_{cls} real web images mined in the current iteration. We set $N_{\text{cls}} = 500$ considering that too many synthetic images may bring in negative effects due to the imperfect logo appearance rendering. Besides, we set logo images of other classes ($j \neq i$) as background scenes for enriching context diversity of class i (Fig. 6). We utilise the SCL synthesising method [7] as in the model bootstrap (Sec. 4.1).

Figure 6: Example logo images with the synthetic context enhancement. Red box: model detection; Green box: synthetic logo ground truth.

Once we have self-mined web training images and generated context enriched synthetic data, we perform detection model fine-tuning at the learning rate of 0.0001 by $6,000 \sim 14,000$ iterations depending on the training data size at each iteration. We adopt the original deep learning loss formulation for both Faster R-CNN and YOLOv2. Model generalisation is expected to improve when the training data quality is sufficient in terms of label accuracy and context richness.

4.4. Incremental Learning Stop Criterion

We conduct incremental model self-co-learning until some stop criterion is met, for example, the model performance gain becomes marginal or zero. We adopt the YOLOv2 as the deployment logo detection model due to its superior efficiency and accuracy (see Table 5). In practice, we can assess the model performance on an independent validation set.

5. Experiments

Competitors We compared the proposed SL^2 model against four types of state-of-the-art object detection methods. **(1)** *Fully supervised object detection*, including a total of five deep learning models (Faster R-CNN [4], SSD [18], YOLOv2 [5], YOLOv3 [20], and RetinaNet [19]). For training, we used the synthetic training data generated by SCL [7], same as SL^2. **(2)** *Weakly supervised object detection*, in particular the Weakly Supervised object Localisation (WSL) [21] and Proposal Cluster Learning (PCL) [22] models, designed for training detectors with image-level class label annotations. Therefore, we can directly utilise the WebLogo-2M data to train a Weakly supervised object detection logo model. Note, noisy logo labels may pose extreme challenges. **(3)** *Webly supervised object detection*, in particular Webly Learning Object Detection (WLOD) [23]. It is a state-of-the-art weakly supervised object detection method where clean Google images are used to train exemplar classifiers which is deployed to classify region proposals by EdgeBox [44]. In our implementation, we further improved the classification component by exploiting an ImageNet and PASCAL trained VGG-16 [45] model as the feature extractor and L2 distance as the matching metric. We adopted the nearest neighbour classification model with the logo icon images (Fig. 4(b)) as labelled data. Additionally, we considered a variant of WLOD by synthesising context enhanced logo icon instances with SCL [7]. **(4)** *Universal logo detection* [29, 14] that collectively treats all logo classes as the positive class. Following [29, 14], we reformulated the original multi-class regional proposal learning into a binary-class version. We used the same synthetic training data as our model.

Performance Metrics To measure logo detection performance, we used the Average Precision (AP) for each individual logo class, and the mean Average Precision (mAP) for all classes [46]. A detection is considered being correct when the Intersection over Union (IoU) between the predicted and groundtruth exceeds 50%.

5.1. Comparative Evaluations

We compared the scalable logo detection performance on the test data of WebLogo-2M in Table 3. It is evident that the proposed SL^2 model significantly outperforms all other alternative methods, e.g. surpassing the best baseline WLOD by 27.6% (46.9%-19.3%) in mAP. SL^2 also surpasses our preliminary model SLST due to joint benefits of self-learning and co-learning. Specifically, we have the following observations:

(1) The weakly supervised learning models, WSL [21] and PCL [22], produce the worst results, due to the joint effects of complex logo appearance variations and large proportions of false positive images (Fig. 2).

(2) The WLOD method performs reasonably well, suggesting that the joint auxiliary knowledge from clean logo icon images and general object data of ImageNet and Pas-

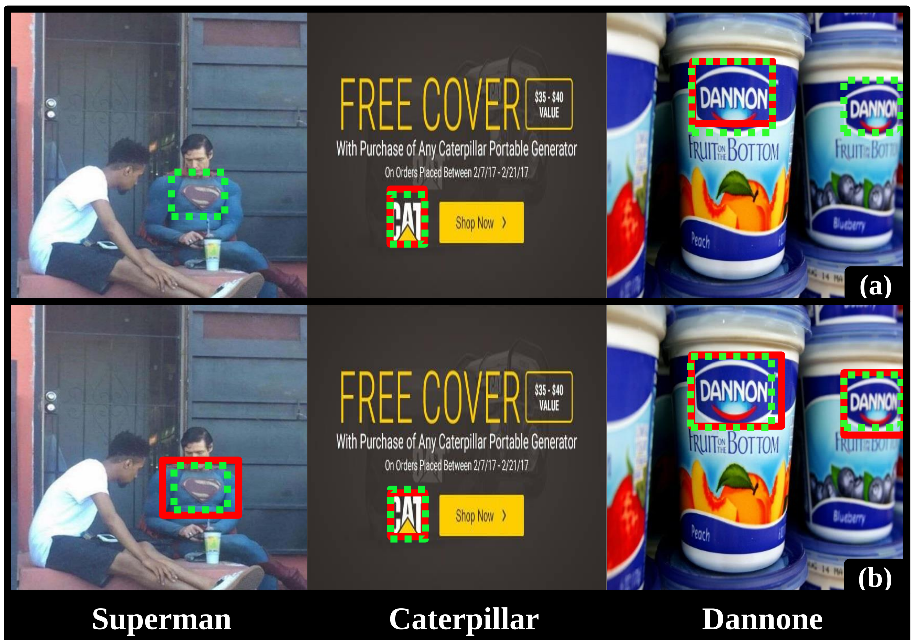

Figure 7: Qualitative evaluations of the **(a)** WLOD and **(b)** SL^2 models. Green dashed boxes: ground truth. Red solid boxes: detected. The WLOD fails to detect visually ambiguous (1st column) logo instance, success on relatively clean (2nd column) logo instances, while only fires partially on the salient one (3rd column). The SL^2 model can correctly detect all these logo instances with varying context and appearance quality.

Table 3: Logo detection performance on WebLogo-2M.

Method	mAP (%)
SSD [18]	8.8
Faster R-CNN [4]	14.9
YOLOv2 [5]	18.4
YOLOv3 [20]	11.0
RetinaNet [19]	4.1
WSL [21]	3.6
PCL [22]	0.2
WLOD [23]	19.3
WLOD [23] + SCL [7]	7.8
ULD [29, 14]	13.2
SLST [24]	36.8
SL^2 (Ours)	**46.9**

cal VOC is transferable.

(3) By using the synthetic training data with rich context, fully supervised detection models YOLOv2 and Faster R-CNN are able to achieve relatively strong results. This suggests that context enhancement is critical for object detection, and the combination of *strongly* supervised learning model + training data synthesising is superior to *weakly* supervised learning. Interestingly, unlike the previous findings [20], it is observed differently that two arguably stronger models YOLOv3 and RetinaNet yield even *weaker* results. We consider that this is due to two reasons: (a) The existence of noisy training labels that bring about more severe harm to methods with more discriminative learning capabilities; (b) A higher sensitivity to the gap between synthetic and real logo images resulted from stronger fitting to potentially noisy training data.

(4) Another supervised one-stage model SSD yields weak detection performance. This is similar to the original finding that SSD is more sensitive to object size with weaker detection performance on small objects as in-the-wild logo instances [18].

(5) WLOD+SCL gives a weaker result (7.8%) than WLOD (19.3%). This indicates that joint supervised learning is critical for exploiting enhanced context.

(6) ULD gives a weaker performance (13.2%) compared to the standard Faster R-CNN (14.9%). This implies that it is not scalable to cases with a large number of logo classes – A multi-class detection learning can already well mine the class agnostic property.

Qualitative Evaluation For visual comparison, we show a number of qualitative logo detection examples from three classes by the SL^2 and WLOD models in Fig. 7.

5.2. Further Analysis and Discussions

5.2.1. Effects of Incremental Model Self-Co-Learning

We evaluated the effects of incremental model self-co-learning on discovered training data and context enriched synthetic images by examining the model performance of SL^2 at individual iterations. Table 4 and Fig. 8 show that SL^2 improves consistently from the 1^{st} to 8^{th} iterations of self-co-learning. In particular, the starting data mining brings about the maximal mAP gain of 10.2% (28.6%-18.4%) with per-iteration benefit dropping gradually. This suggests that our model design is capable of effectively addressing the notorious error propagation challenge thanks to (1) a proper detection model initialisation by logo context synthesising for providing a sufficiently good starting-point detection; (2) a strict selection on self-evaluated detections for reducing the amount of false positives and suppressing the likelihood of error propagation; and (3) cross-model co-learning with cross-class context enhancement with the capability of addressing the class imbalanced data learning problem whilst enhancing the model robustness against unconstrained background. We also observed that more images are mined along the process, indicating that SL^2 effectively improves over time in the capability of tackling more complex context. However, false positives with similar/confusing appearance can be inevitably introduced during automated self-discovery of new training data in the iterative learning process, causing failure cases during model inference (Fig. 9).

Table 4: Model performance development over incremental SL^2 iterations.

Iteration	mAP	mAP Gain	Training Images
0	18.4	N/A	5,862
1^{st}	28.6	**10.2**	21,610
2^{nd}	33.2	4.6	41,314
3^{rd}	39.1	5.9	54,387
4^{th}	42.2	3.1	74,855
5^{th}	44.4	2.2	86,599
6^{th}	45.6	1.2	98,055
7^{th}	**46.9**	1.3	**107,327**
8^{th}	**46.9**	0.0	**Stop**

Table 5: Co-learning *versus* self-learning.

Method	mAP (%)
Self-Learning (Faster R-CNN)	36.8
Self-Learning (YOLO)	39.4
Co-Learning (Faster R-CNN)	44.2
Co-Learning (YOLO) (**SL**^{2})	**46.9**

5.2.2. Effects of Cross-Model Co-Learning

We assessed the benefits of cross-model co-learning between Faster R-CNN and YOLOv2 in SL^2 in comparison of the single-model *self-learning* strategy. In contrast to co-learning, the self-learning exploits self-mined new training data for incremental model update without the benefit of cross-model complementary advantages. Table 5 and Fig. 8 show that both models benefit clear performance gains from co-learning, e.g. 7.4% (44.2-36.8) for Faster R-CNN, and 7.5% (46.9-39.4) for YOLOv2. This verifies our motivation of exploiting the co-learning principle for maximising the complementary advantages of distinct model formulations in the scalable logo model optimisation.

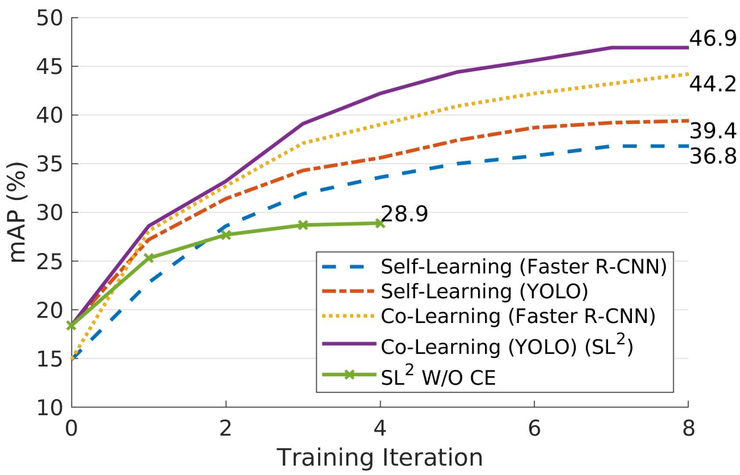

Figure 8: Evaluating the model co-learning and self-learning strategies, and the effect of Context Enhancement (CE) based training data class balancing.

5.2.3. Effects of Synthetic Context Enhancement

We evaluated the impact of context enhancement (i.e. the cross-class context enriched synthetic training data) on model performance. Table 6 shows that context enhancement not only provides a clear model improvement across iterations due to the suppression of negative imbalance learning effect, but also simultaneously enlarges the data mining capacity due to potentially less noisy training data aggregation. Without context enhancement and training class balancing, the model stops to improve by the 4^{th} learning iteration, resulting in weaker performance at 28.9% vs. 46.9% by the full SL^2 model. This verifies the importance of context enhancement and class balancing for detection model learning, validating our model design considerations.

Table 6: Effects of training data Context Enhancement (CE). Metric: mAP (%).

Iteration	0	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
With CE	18.4	**28.6**	**33.2**	**39.1**	**42.2**	**44.4**
Without CE	18.4	25.3	27.7	28.7	28.9	28.0

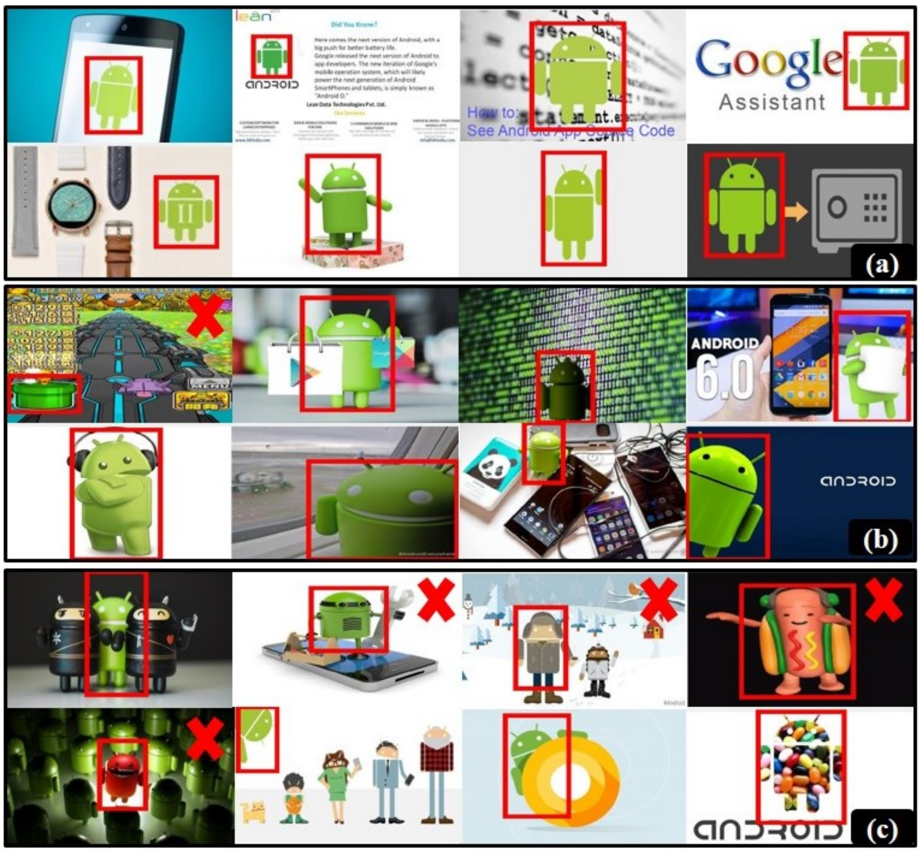

Figure 9: Randomly selected images self-discovered in the **(a)** 1^{st}, **(b)** 4^{th}, and **(c)** 8^{th} iterations for the logo class "Android". Red box: SL^2 model detection. Red cross: false detection. The images mined in the 1^{st} iteration have clean logo instances and background, whilst those discovered in the 4^{th} and 8^{th} iterations have more diverse logo appearance variations in richer and more complex context. More false positives are likely to be produced in the 4^{th} and 8^{th} self-discovery.

5.2.4. *Estimating the Bootstrap Synthetic Data Size*

For efficiency, we estimated the synthetic data size in model bootstrap with YOLOv2. Table 7 shows that whilst more synthetic training data generally lead to higher mAP rates, the benefit is rapidly diminishing with size increasing. Besides, this gain comes with drastically higher model training cost. According to the resource limit, we generated 1,000 synthetic images per class in our main experiments.

Table 7: Estimating the bootstrap synthetic data size using YOLOv2.

Number of Images Per Class	mAP (%)
100	15.6
300	17.2
1,000	**18.4**

6. Conclusion and Future Work

In this work, we presented a scalable logo detection method including dataset establishment and model learning. This is realised by exploring the web data learning principle without a tedious need of manually labelling fine-grained logo bounding boxes. Specifically, we proposed a new incremental learning method named *Scalable Logo Self-co-Learning* (SL^2). It uniquely enables reliable self-discovery and auto-labelling of new training images from unconstrained in-the-wild web data to progressively improve the model detection capability in a cross-model co-learning manner. We constructed a very large logo benchmark WebLogo-2M by automatically collecting and processing free web data in a scalable manner. This facilitates the community for further investigation of scalable logo detection in the future. We have conducted extensive comparative evaluations and analysis on the benefits of incremental model training and context enhancement on the WebLogo-2M benchmark. The results show the advantages and superiority of our SL^2 method over the state-of-the-art alternative methods, ranging from strongly-supervised and weakly-supervised detection models to webly learning models. We finally provided in-depth model component analysis and evaluations for giving insights on model performance gain and formulation.

As an early attempt for scalable logo detection in deep learning, our approach still has a number of limitations that need be addressed in the future work. *First*, the web imagery data we collected are over noisy, imposing an extreme challenge for data selection during self-labelling. Therefore, developing superior data collection is one of the most effective methods. *Second*, the proposed SL^2 model relies heavily on the detection scores of object instances which is error prone partly due to the model over-confident on unknown classes. How to mitigate this effect is worth more investigation. *Third*, the detection models we leveraged in designing SL^2 are not sufficiently efficient to process millions of images. An important future research is to develop more cost-effective object detection models. We reckon that with dedicated development in the above directions, the scalability of logo detection can be advanced significantly.

Acknowledgement

This work was partially supported by the China Scholarship Council, Vision Semantics Limited, the Royal Society Newton Advanced Fellowship Programme (NA150459), Innovate UK Industrial Challenge Project on Developing and Commercialising Intelligent Video Analytics Solutions for Public Safety (98111-571149), and the Alan Turing Institute Fellowship Project on Deep Learning for Large-Scale Video Semantic Search.

References

[1] T. D. Pham, Unconstrained logo detection in document images, Pattern recognition 36 (12) (2003) 3023–3025.

[2] C. Pan, Z. Yan, X. Xu, M. Sun, J. Shao, D. Wu, Vehicle logo recognition based on deep learning architecture in video surveillance for intelligent traffic system, in: IET International Conference on Smart and Sustainable City, 2013, pp. 123–126.

[3] K.-W. Li, S.-Y. Chen, S. Su, D.-J. Duh, H. Zhang, S. Li, Logo detection with extendibility and discrimination, Multimedia tools and applications 72 (2) (2014) 1285–1310.

[4] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.

[5] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115 (3) (2015) 211–252.

[7] H. Su, X. Zhu, S. Gong, Deep learning logo detection with data expansion by synthesising context, in: IEEE Winter Conference on Applications of Computer Vision, IEEE, 2017, pp. 530–539.

[8] Y. Liao, X. Lu, C. Zhang, Y. Wang, Z. Tang, Mutual enhancement for detection of multiple logos in sports videos, in: IEEE International Conference on Computer Vision, 2017, pp. 4856–4865.

[9] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, Y. Avrithis, Scalable triangulation-based logo recognition, in: ACM International Conference on Multimedia Retrieval, 2011, p. 20.

[10] S. Romberg, L. G. Pueyo, R. Lienhart, R. Van Zwol, Scalable logo recognition in real-world images, in: Proceedings of the 1st ACM International Conference on Multimedia Retrieval, ACM, 2011, p. 25.

[11] Y. Li, Q. Shi, J. Deng, F. Su, Graphic logo detection with deep region-based convolutional networks, in: IEEE Visual Communications and Image Processing, 2017, pp. 1–4.

[12] A. Joly, O. Buisson, Logo retrieval with a contrario visual query expansion, in: ACM International Conference on Multimedia, 2009, pp. 581–584.

[13] S. C. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, Q. Wu, Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks, arXiv preprint arXiv:1511.02462.

[14] A. Tzk., C. Herrmann., D. Manger., J. Beyerer., Open set logo detection and retrieval, in: Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Vol. 5, 2018, pp. 284–292.

[15] H. Sahbi, L. Ballan, G. Serra, A. Del Bimbo, Context-dependent logo matching and recognition, IEEE Transactions on Image Processing 22 (3) (2012) 1018–1031.

[16] C. H. Lampert, H. Nickisch, S. Harmeling, Attribute-based classification for zero-shot visual object categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (3) (2014) 453–465.

[17] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., Devise: A deep visual-semantic embedding model, in: Advances in neural information processing systems, 2013, pp. 2121–2129.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[19] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.

[20] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767.

[21] D. Li, J.-B. Huang, Y. Li, S. Wang, M.-H. Yang, Weakly supervised object localization with progressive domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3512–3520.

[22] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu, A. L. Yuille, Pcl: Proposal cluster learning for weakly supervised object detection, IEEE transactions on pattern analysis and machine intelligence (2018) early access 1–1.

[23] X. Chen, A. Gupta, Webly supervised learning of convolutional networks, in: IEEE International Conference on Computer Vision, 2015, pp. 1431–1439.

[24] H. Su, S. Gong, X. Zhu, Weblogo-2m: Scalable logo detection by deep learning from the web, in: IEEE International Conference on Computer Vision Workshops, 2017, pp. 270–279.

[25] A. P. Psyllos, C.-N. E. Anagnostopoulos, E. Kayafas, Vehicle logo recognition using a sift-based enhanced matching scheme, IEEE Transactions on Intelligent Transportation Systems 11 (2) (2010) 322–328.

[26] S. Romberg, R. Lienhart, Bundle min-hashing for logo recognition, in: Proceedings of the 3rd ACM conference on International conference on multimedia retrieval, ACM, 2013, pp. 113–120.

[27] L. Nanni, S. Ghidoni, S. Brahnam, Handcrafted vs. non-handcrafted features for computer vision classification, Pattern Recognition 71 (2017) 158–172.

[28] D. M. Montserrat, Q. Lin, J. Allebach, E. J. Delp, Logo detection and recognition with synthetic images, Electronic Imaging 2018 (10) (2018) 337–1.

[29] I. Fehérvári, S. Appalaraju, Scalable logo recognition using proxies, in: IEEE Winter Conference on Applications of Computer Vision, IEEE, 2019, pp. 715–725.

[30] K. Nigam, R. Ghani, Analyzing the effectiveness and applicability of co-training, in: Proceedings of the ninth international conference on Information and knowledge management, 2000, pp. 86–93.

[31] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: Seventh IEEE Workshop on Applications of Computer Vision, 2005.

[32] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the eleventh annual conference on Computational learning theory, ACM, 1998, pp. 92–100.

[33] W. Wang, Z.-H. Zhou, Analyzing co-training style algorithms, in: European Conference on Machine Learning, Springer, 2007, pp. 454–465.

[34] Z. Jiang, S. Zhang, J. Zeng, A hybrid generative/discriminative method for semi-supervised classification, Knowledge-Based Systems 37 (2013) 137–145.

[35] A. Appice, P. Guccione, D. Malerba, A novel spectral-spatial co-training algorithm for the transductive classification of hyperspectral imagery data, Pattern Recognition 63 (2017) 229–245.

[36] J. Tao, D. Zhou, F. Liu, B. Zhu, Latent multi-feature co-regression for visual recognition by discriminatively leveraging multi-source models, Pattern Recognition 87 (2019) 296–316.

[37] O. M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, in: British Machine Vision Conference, Vol. 1, 2015, p. 6.

[38] H. He, E. A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[39] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes challenge: A retrospective, International Journal of Computer Vision 111 (1) (2015) 98–136.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[41] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, K. Saenko, Lsda: Large scale detection through adaptation, in: Advances in Neural Information Processing Systems, 2014, pp. 3536–3544.

[42] R. G. Cinbis, J. Verbeek, C. Schmid, Weakly supervised object localization with multi-fold multiple instance learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (1) (2017) 189–203.

[43] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, J. Xiao, Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop, arXiv preprint arXiv:1506.03365.

[44] C. L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: European conference on computer vision, Springer, 2014, pp. 391–405.

[45] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[46] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2) (2010) 303–338.