WebLogo-2M: Scalable Logo Detection by Deep Learning from the Web

Hang Su
Queen Mary University of London
hang.su@qmul.ac.uk

Shaogang Gong
Queen Mary University of London
s.gong@qmul.ac.uk

Xiatian Zhu
Vision Semantics Ltd.
eddy@visionsemantics.com

Abstract

Existing logo detection methods usually consider a small number of logo classes and limited images per class with a strong assumption of requiring tedious object bounding box annotations, therefore not scalable to real-world applications. In this work, we tackle these challenges by exploring the webly data learning principle without the need for exhaustive manual labelling. Specifically, we propose a novel incremental learning approach, called Scalable Logo Self-Training (SLST), capable of automatically self-discovering informative training images from noisy web data for progressively improving model capability. Moreover, we introduce a very large (1,867,177 images of 194 logo classes) logo dataset "WebLogo-2M"[1] *by an automatic web data collection and processing method. Extensive comparative evaluations demonstrate the superiority of the proposed SLST method over state-of-the-art strongly and weakly supervised detection models and contemporary webly data learning alternatives.*

1. Introduction

Automated logo detection from "in-the-wild" (unconstrained) images benefits a wide range of applications in many domains, e.g. brand trend prediction for commercial research and vehicle logo recognition for intelligent transportation [27, 26, 21]. This is inherently a challenging task due to the presence of many logos in diverse context with uncontrolled illumination, low-resolution, and background clutter (Fig. 1). Existing methods typically consider a small number of logo images and classes under the assumption of having large sized training data annotated at the logo object instance level, i.e. object bounding boxes [14, 15, 27, 25, 26, 1, 17, 21]. Whilst this controlled setting allows a straightforward adoption of state-of-the-art detection models [24, 8], it is unscalable to real-world logo detection tasks when a much larger number of logo classes are of interest but limited by (1) the extremely high cost for constructing therefore unavailability of large scale logo dataset with exhaustive logo instance bounding box labelling [29]; and (2) lacking incremental model learning to progressively update and expand the model to increasingly more training data without such fine-grained labelling. Existing models are mostly one-pass trained and blindly generalised to new test data.

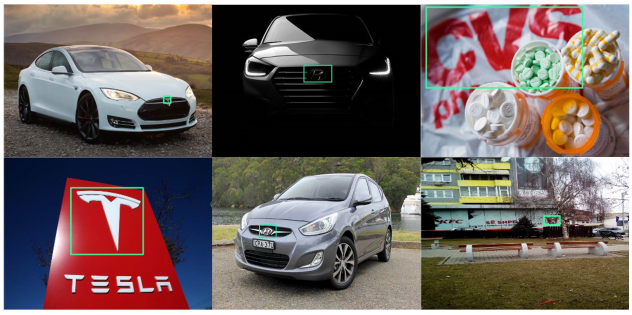

Figure 1: Illustration of logo detection challenges: significant logo variation in object size, illumination, background clutter, and occlusion.

In this work, we consider scalable logo detection in very large collections of unconstrained images without exhaustive fine-grained object instance level labelling for model training. Given that all existing datasets only have small numbers of logo classes, one possible strategy is to learning from a small set of labelled training classes and adopting the model to other novel (test) logo classes, that is, Zero-Shot Learning (ZSL) [33, 16, 7]. 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 embedding space of word vectors [7]. However, they are limited if at all shared attributes or other forms of semantic representations among logos due to their unique characteristics. A lack of large scale logo datasets (Table 1), in both class numbers and image instance numbers per class, limit severely learning scalable logo models. This study explores the webly data learning principle for addressing both large scale dataset construction and incremental logo model learning without exhaustive manual labelling of increasing data expansion. We call this setting *scalable logo detection*.

Our contributions in this work are three-fold: **(1)** We

[1]The WebLogo-2M dataset is available at http://www.eecs.qmul.ac.uk/~hs308/WebLogo-2M.html

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

Dataset	Logos	Images	Supervision	Noisy	Construction	Scalability	Availability
BelgaLogos [14]	37	10,000	Object-Level	✗	Manually	Weak	✓
FlickrLogos-27 [15]	27	1,080	Object-Level	✗	Manually	Weak	✓
FlickrLogos-32 [27]	32	8,240	Object-Level	✗	Manually	Weak	✓
TopLogo-10 [32]	10	700	Object-Level	✗	Manually	Weak	✓
LOGO-NET [12]	160	73,414	Object-Level	✗	Manually	Weak	✗
WebLogo-2M (Ours)	194	1,867,177	Image-Level	✓	Automatically	Strong	✓(Soon)

investigate the scalable logo detection problem, characterised by modelling a large quantity of logo classes *without* exhaustive bounding box labelling. This is significantly under-studied in the literature. **(2)** We propose a novel incremental learning approach to scalable logo detection by exploiting multi-class detection with synthetic context augmentation. We call our method as *Scalable Logo Self-Training* (SLST), since it automatically discovers potential positive logo images from noisy web data to progressively improve the model generalisation in an iterative self-learning manner. **(3)** We introduce a large logo detection dataset with 194 logo classes and 1,867,177 images, called *WebLogo-2M*, by *automatically* sampling webly logo images from the social media Twitter. Importantly, this scheme allows to further expand easily our dataset with new logo classes, and therefore offering a scalable solution for dataset construction. Extensive comparative experiments demonstrate the superiority of the proposed SLST method over not only state-of-the-art strongly (Faster R-CNN [24], SSD [19]) and weakly (WSL [4]) supervised detection models but also webly learning methods (WLOD [2]), on the newly introduced WebLogo-2M dataset .

2. Related Works

Logo Detection Early logo detection methods are established on hand-crafted visual features (e.g. SIFT and HOG) and conventional classification models (e.g. SVM) [17, 25, 26, 1, 15]. In these methods, only small logo datasets are evaluated with a limited number of both logo images and classes modelled. A few deep methods [13, 12, 32] have been recently proposed by exploiting the state-of-the-art object detection models such as R-CNN [9, 24, 8]. This in turn inspires large data construction [12]. However, all these existing models are not scalable to real world deployments 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 not scalable to a realistically large number of logo classes given limited human labelling effort. In contrast, our method eliminates both needs by allowing the detection model learning from image-level weakly annotated and noisy images automatically collected from the social media (webly). As such, we enable automated introduction of any quantity of new logos for both dataset construction/expansion and model updating without the need for exhaustive manual labelling.

Logo Datasets A number of logo benchmark datasets exist (Table 1). Most existing datasets are constructed *manually* and typically small in both image number and logo category thus insufficient for deep learning. Recently, Hoi et al. [12] attempt to address this small logo dataset problem by creating a large LOGO-NET dataset. However, this dataset is not publicly accessible. To address this scalability problem, we propose to collect logo images *automatically* from the social media. This brings 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 logo categories. To our knowledge, this is the first attempt to construct a large scale logo dataset by exploiting inherently noisy web data.

Self-Training Self-training is a special type of incremental learning wherein 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 [20]. A similar approach to our model is the detection model by Rosenberg et al. [28]. This model also explores the self-training mechanism. However, this method needs a number of per class strongly and accurately labelled training data at the object instance level to initialise their detection model. Moreover, it assumes all unlabelled images belong to the target object categories. These two assumptions limit severely model effectiveness and scalability given webly collected training data without any object bounding box labelling whilst with a high ratio of noisy irrelevant images.

3. WebLogo-2M Logo Detection Dataset

We present a scalable method to automatically construct a large logo detection dataset, called *WebLogo-2M*, with 1,867,177 webly images from 194 logo classes (Table 2).

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

Logos	Raw Images	Filtered Images	Noise Rate (%)
194	4,047,129	1,867,177	Varying
-	-	(5/2583/141,480)	(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 the 32 logo classes of FlickrLogo-32 [27] and the 10 logo classes of TopLogo-10 [32]. Specifically, the logo class selection was guided by an extensive review of social media reports regarding to the 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 and more critically, unlimited data access permission therefore facilitating the collection of large scale logo images[7].

Image Collection We collected 4,047,129 webly logo images. Specifically, through the Twitter API, one can automatically retrieve images from tweets by matching query keywords against tweets in real time. In our case, we query the logo brand names so that images in tweets containing the query words can be extracted. The retrieved 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 1,867,177 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 (noisy). (2) *Duplicate Removal*: We identified and discarded exact-duplicates (i.e. multiple copies of the same image). Specifically, given an reference image, we removed those with identical width and height. This image spacial size based scheme is not only computationally cheaper than the appearance based alternative [22], but also very 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.

[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

[7] We also attempted at Google and Bing search engines, and three other social media (Facebook, Instagram, and Flickr). However, all of them are rather restricted in data access and limiting incremental big data collection, e.g. Instagram allows only 500 times of image downloading per hour through the official web API.

3.2. Properties of WebLogo-2M

Compared to existing logo detection databases [14, 27, 12, 32], this webly logo image dataset presents three unique 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 by the query keywords. These labels are obtained automatically in data collection without human fine-grained labelling. This is much more scalable than manually annotating accurate individual logo bounding boxes, particularly when the number of both logo images and classes are very large.

(II) Noisy (False Positives) Images collected from online web sources are inherently noisy, e.g. often no logo objects appearing in the images therefore providing plenty of natural false positive samples. For estimating a degree of noisiness, we sampled randomly 1,000 web images per class for all 194 classes and manually examined whether they are true or false logo images[8]. As shown in Fig. 2, the true logo image ratio varies significantly among 194 logos, e.g. **75%** for "Rittersport" vs. **0.2%** for "3M". On average, true logo images take only 21.26% vs. the remaining as false positives. Such noisy images pose extremely high challenges to model learning, even though there are plenty of data scalable to very large size in both class numbers and samples per class.

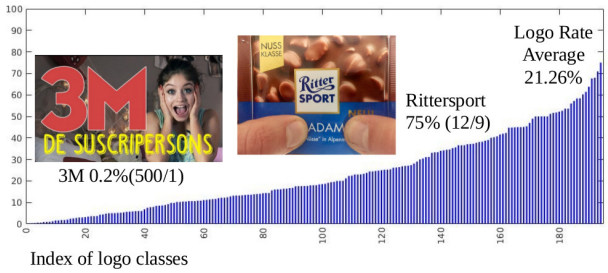

Figure 2: True logo image ratios (%). This was estimated from 1,000 random logo images per class over 194 classes.

(III) Class Imbalance The WebLogo-2M dataset presents a natural logo object occurrence imbalance in daily 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 causes extra labelling effort but also renders the model learning challenges *unrealistic*. In contrast, we preserve the inherent class imbalance nature in the data for fully automated dataset construction and retaining more realistic data for model learning, which requires minimising model learning bias towards densely-sampled classes [10].

[8] In the case of sparse logo classes with less than 1,000 webly collected images, we examined all available images.

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 bounding boxes only for facilitating viewing. Most images contain no "Adidas" object, i.e. false positives, suggesting a high noise degree in webly collected data. **(b)** Clean images of 194 logo classes automatically collected from the Google Image Search, used in synthetic training images generation and augmentation. **(c)** One example true positive webly (Twitter) image per logo class, totally 194 images, showing the rich and diverse context in unconstrained images where typical logo objects reside in reality.

Further Remark Since the proposed dataset construction method is completely automated, new logo classes can be easily added without human labelling. This permits good scalability to enlarging the dataset cumulatively, in contrast to existing methods [29, 12, 18, 5, 14, 27, 12, 32] that require exhaustive human labelling therefore hampering further dataset updating and enlarging. This automation is particularly important for creating object detection datasets with expensive needs for labelling explicitly object bounding boxes, than building cheaper image-level class annotation datasets [11]. While being more scalable, this WebLogo-2M dataset also provides more realistic challenges for model learning given weaker label information, noisy image data, unknown scene context, and significant

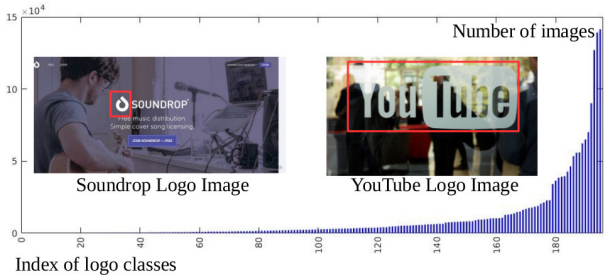

Figure 3: Imbalanced logo image class distribution, ranging from 3 images ("Soundrop") to 141,412 images ("Youtube"), i.e. 47,137 imbalance ratio.

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 (1,867,177 images) as the *training* data. For performance evaluation, a set of images with fine-grained object-level annotation groundtruth is required. To that end, we construct an independent *test set* of 6,019 logo images with logo bounding box labels by (1) assembling 2,870 labelled images from the FlickrLogo-32 [27] and TopLogo [32] datasets and (2) manually labelling 3,149 images independently collected from the Twitter. Note that, the only purpose of labelling this test set is for performance evaluations of different detection methods, independent of WebLogo-2M construction.

4. Self-Training A Multi-Class Logo Detector

We aim to automatically train a multi-class logo detection model incrementally from noisy and weakly labelled web images. Different from existing methods building a detector in a one-pass "batch" procedure, we propose to incrementally enhance the model capability "sequentially", in the spirit of self-training [20]. This is due to the *unavailability* of sufficient accurate fine-grained training images per class. In other words, the model must self-select trustworthy images from the noisy webly labelled data (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 which in turn produces error-prone predictions. This may cause *model drift* – the errors in model prediction will be propagated through the iterations therefore have the potential to corrupt the model knowledge structure. Also, the inherent data imbalance over different logo classes may make model learning biased towards only a few number of majority classes, therefore leading to significantly weaker capability in detecting minority classes. Moreover, the two problems above are intrinsically interdependent with one possibly negatively affecting the other. It is non-trivial to solve these challenges without exhaustive fine-grained human annotations.

Rational of Model Design In this work, we present a scalable logo detection learning solution capable of addressing the aforementioned two issues in a self-training framework. The intuition is: Web knowledge provides ambiguous but still useful coarse image level logo annotations, whilst self-training offers a scalable learning means to explore iteratively such weak information. We call our method Scalable Logo Self-Training (SLST). In SLST, we select strongly-supervised rather than weakly-supervised baseline models to initialise the self-training process for two reasons: (1) The performance of weakly-supervised models are much inferior than that of strongly supervised counterparts [3]; (2) The noisy webly weak labels may further hamper the effectiveness of weakly supervised learning. A schematic overview of the entire SLST process is depicted in Fig. 5.

4.1. Model Bootstrap

To start SLST, we first need to provide a reasonably discriminative logo detection baseline model with sufficient bootstrapping training data discovery. In our implementation, we choose the Faster R-CNN [24] due to its good performance on detecting varying-size objects [32]. Other alternatives e.g. SSD [19] and YOLO [23] can be readily integrated. The choice of this baseline model is independent of the proposed SLST method. Faster R-CNN needs strongly supervised learning from object-level bounding box annotations to gain detection discrimination, which however is not available in our scalable webly learning setting.

To overcome this problem, we propose to exploit the idea of synthesising fine-grained training logo images, therefore maintaining model learning scalability for accommodating large quantity of logo classes. In particular, this is achieved by generating synthetic training images as in [32]: 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 the Google Image Search by querying the corresponding logo class name (Fig. 4 (b)). The background images can be chosen flexibly, e.g. the non-logo images in the FlickrLogo-32 dataset [27] or others retrieved by irrelevant query words from web search engines. To enhance appearance variations in synthetic logos, colour and geometric transformation can be applied [32].

Training Details We synthesised 100 training images per logo class, in total 19,400 images. For learning the Faster R-CNN, we set the learning rate 0.0001, and the learning iterations $6,000$ to $14,000$ depending on the training data size at each iteration. Following [32], we pre-trained the detector on ImageNet object classification images [29] for model warmup.

Figure 5: Overview of the Scalable Logo Self-Training (SLST) method. **(1)** Model initialisation by using synthetic logo training images (Sec. 4.1). **(2)** Incrementally self-mining positive logo images from noisy web data pool (Sec. 4.2). **(3)** Balance training data by synthetic context augmentation on mined data (Sec. 4.3). **(4)** Using both mined web images and context-enhanced synthetic images for model updating (Sec. 4.4). This process is repeated iteratively for progressive training data mining and model update.

4.2. Incremental Self-Mining Noisy Web Images

After the logo detector is discriminatively bootstrapped, we proceed to improve its detection capability by incrementally self-mining potentially positive logo images from weakly labelled WebLogo-2M data. To identify the most compatible training images, we define a selection function using the detection score of up-to-date model:

$$S(\mathcal{M}_t, \boldsymbol{x}, y) = S_{\text{det}}(y|\mathcal{M}_t, \boldsymbol{x}) \quad (1)$$

where $\mathcal{M}_t$ denotes the t-th step detector model, and $\boldsymbol{x}$ denotes a logo image with the web image-level label $y \in Y = \{1, 2, \cdots, m\}$ with m the total logo class number. $S_{\text{det}}(y|\mathcal{M}_t, \boldsymbol{x}) \in [0, 1]$, indicates the maximal detection score of $\boldsymbol{x}$ on the logo class y by model $\mathcal{M}_t$. For positive logo image selection, we need a high threshold detection confidence (0.9 in our experiments) [35] for strictly controlling the impact of model detection errors in degrading the incremental learning benefits. This new training data discovery process is summarised in Alg. 1.

4.3. Cross-Class Synthetic Context Augmentation

Inspired by the benefits of context enhancement in logo detection [32], we propose the idea of cross-class context augmentation for not only fully exploring the contextual richness of WebLogo-2M data but also addressing the intrinsic imbalanced logo class problem where model learning is likely biased towards well-labelled classes (the majority classes) resulting in poor performance against sparsely-labelled classes (the minority classes) [10].

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

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

Algorithm 1 Incremental Self-Mining Noisy Web Images

Input: Current model $\mathcal{M}_{t-1}$, Unexplored 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 i as a potential positive logo image;
    if Meeting selection criterion (Eq. (1))
      $\mathcal{T}_t = \mathcal{T}_t \cup \{i\}$;
      $\mathcal{D}_t = \mathcal{D}_t \setminus \{i\}$;
    end if
end for

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

Therefore, we only perform synthetic data augmentation for those classes with only $<N_{\text{cls}}$ 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 against background. Importantly, we choose the self-mined logo images of other classes ($j \neq i$) as the background images for particularly enriching the contextual diversity for improving logo class i (Fig. 6). We utilise the SCL synthesising method [32] as in model bootstrap (Sec. 4.1).

4.4. Model Update

Once we have self-mined and context enriched synthetic training data, we incrementally update the detection model by fine-tuning batch-wise training. Model generalisation is

Figure 6: Example images by synthetic context augmentation. Red box: model detection; Green box: synthetic logo.

to be improved when the new training data quality is sufficient in terms of both true positives percentage and the context richness.

5. Experiments

Competitors We compared the proposed SLST model with five state-of-the-art alternative detection approaches: **(1)** Faster R-CNN [24]: A competitive region proposal driven object detection model which is characterised by jointly learning region proposal generation and object classification in a single deep model In our scalable webly learning context, the Faster R-CNN is optimised with synthetic training data generated by the SCL [32] method, exactly the same as our SLST model. **(2)** SSD [19]: A state-of-the-art regression optimisation based object detection model. We similarly learn this strongly supervised model with synthetic logo instance bounding box labels as Faster R-CNN above. **(3)** Weakly Supervised object Localisation (WSL) [4]: A state-of-the-art weakly supervised detection model allowing to be trained with image-level logo label annotations in a multi-instance learning framework. Therefore, we can directly utilise the webly labelled WebLogo-2M images to train the WSL detection model. Note that, noisy logo labels inherent to web data may pose additional challenges in addition to high complexity in logo appearance and context. **(4)** Webly Learning Object Detection (WLOD) [2]: 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 [36]. In our implementation, we further improved the classification component by exploiting an VGG-16 [31] model trained by the ImageNet-1K & Pascal VOC data as a stronger feature extractor and the L2 distance as the matching metric. We adopted the nearest neighbour classification model with Google logo images (Fig. 4(b)) as the labelled training data. **(5)** WLOD+SCL: a variant of WLOD [2] with context enriched training data by exploiting SCL [32] to synthesise various context for Google logo images.

Performance Metrics For the quantitative performance measure of logo detection, we utilised the Average Precision (AP) for each individual logo class, and the mean Average Precision (mAP) for all classes [6]. A detection is considered corrected when the Intersection over Union (IoU) between the predicted and groundtruth exceeds 50%.

Table 3: Logo detection performance comparison.

Model	mAP (%)
Faster R-CNN [24]	14.59
SSD [19]	9.02
WSL [4]	4.28
WLOD [2]	17.35
WLOD[2] + SCL[32]	7.72
SLST	**34.37**

5.1. Comparative Evaluations

We compared the logo detection performance on the WebLogo-2M benchmarking test data in Table 3. It is evident that the proposed SLST model significantly outperforms all other alternative methods, e.g. surpassing the best baseline WLOD by 17.02% (34.37%-17.35%) in mAP. We also have the following observations: **(1)** The weakly supervised learning based model WSL produces the worst result, due to the joint effects of complex logo appearance variation against unconstrained context and high proportions of false positive logo images (Fig. 2). **(2)** WLOD method performs reasonably well suggesting that the knowledge learned from auxiliary data sources (ImageNet and Pascal VOC) is transferable to some degree, confirming the similar findings as in [30, 34]. **(3)** By utilising synthetic training images with rich context and background, fully supervised model Faster R-CNN is able to achieve the 3rd best results among all competitors. This suggests that context augmentation is critical for object detection model optimisation, and the combination of *strongly* supervised learning model + auto training data synthesising is a preferred strategy over *weakly* supervised learning in webly learning setting. The regression detection model SSD yields lower performance. One plausible reason is the inherent weaker capability of non-proposal detection model in locating small objects such as in-the-wild logo instances (Fig. 1). **(4)** Interestingly, WLOD + SCL produces a weaker result (7.72%) compared to WLOD (17.35%) suggesting that joint supervised learning is critical to exploit context enriched data augmentation, otherwise introducing distracting effects resulting in degraded matching. For visual comparison, qualitative evaluations for SLST and WLOD are shown in Fig. 7.

5.2. Further Analysis and Discussions

Effects of Incremental Model Self-Training We evaluated the effects of incremental learning on self-discovered training data and context enriched synthetic images by examining the SLST model performance at individual iterations. Table 4 shows that the SLST model improves consistently over iterations of self-training [9], with the starting data mining bringing in the maximal mAP gain 8.00% (22.59%-

[9]We stopped after four rounds of self-training since the obtained performance gain is not significant.

Figure 7: Quantitative evaluations of the **(a)** WLOD and **(b)** SLST models. Red box: detected. Green box: ground truth. WLOD fails to detect visually ambiguous (1st column) and small-sized (2nd column) logo instances, while only fires partially on the salient one (3rd column). The SLST model can correctly detect all these logo instances with varying context and appearance quality.

Table 4: Effects of incremental model self-training in SLST.

Iteration	0	1	2	3	4
mAP (%)	14.59	22.59	28.85	31.86	**34.37**
Gain (%)	N/A	**8.00**	6.26	3.01	2.51
Mined Image	4,235	23,615	47,183	76,643	**95,722**

14.59%) and the 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 sufficient starting detection; (2) a strict selection on self-evaluated detections for reducing the amount of false positives, suppressing the likelihood of error propagation; and (3) the cross-logo context enriched synthetic training data augmentation and balancing for addressing the imbalanced data learning problem whilst enhancing the model robustness against diverse unconstrained background clutters. We also observed that more images are mined along the incremental data mining process, suggesting that the SLST model improves over time in the capability of tackling more complex context, although potentially simultaneously leading to more false positives which can cause lower model growing rates, as indicated in Fig. 8.

Effects of Synthetic Context Enhancement We evaluated the impact of training data context enhancement (i.e. the cross-class context enriched synthetic training data) on the SLST model performance. Table 5 shows that context augmentation brings in 4.87% (34.37%-29.50%) mAP improvement. This suggests the importance of context and data balance in detection model learning, confirming our model design intuition.

6. Conclusion

We present a scalable end-to-end logo detection solution including logo dataset establishment and multi-class logo detection model learning, realised by exploring the webly data learning principle without the cost of manually labelling fine-grained logo annotations. Particularly, we propose a new incremental learning method named Scalable Logo Self-Training (SLST) for enabling reliable self-discovery and auto-labelling of new training images from noisy web data to progressively improve the model detection capability in unconstrained in-the-wild images. Moreover, we construct a very large logo detection benchmarking dataset WebLogo-2M by automatically collecting and processing web stream data (Twitter) in a scalable manner, therefore facilitating and motivating the further investigation of scalable logo detection in the near future. We have validated the advantages and superiority of the proposed SLST approach in comparisons to state-of-the-art alternative methods ranging from strongly- and weakly-supervised detection models to webly data learning models through extensive comparative evaluations and analysis on the benefits of incremental model training and context enhancement, using the newly introduced WebLogo-2M logo benchmark dataset.

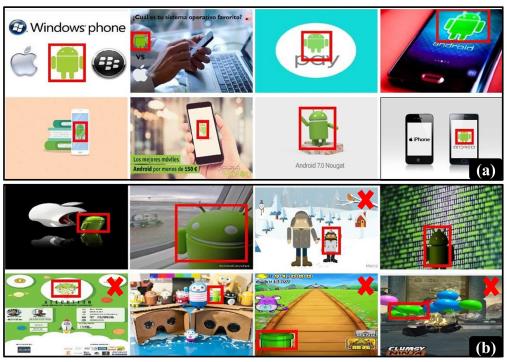

Figure 8: Randomly selected images self-discovered in the **(a)** 1st and **(b)** 4th iteration for the logo class "Android". Red box: SLST model detection. Red cross: false detection. The images mined in the 1st iteration have clean logo instances and background, whilst those discovered in the 4th iteration have more varied and ambiguous logo instances in more complex context. More false detections are produced in the 4th self-discovery.

Table 5: Effects of training data Context Enhancement (CE) on SLST self-training. Metric: mAP (%).

CE	0	1	2	3	4
✗	14.59	17.44	24.34	27.81	29.50
✓	14.59	**22.59**	**28.85**	**31.86**	**34.37**

Acknowledgements

This work was partially supported by the China Scholarship Council, Vision Semantics Ltd., and the Royal Society Newton Advanced Fellowship Programme (NA150459).

References

[1] R. Boia, A. Bandrabur, and C. Florea. Local description using multi-scale complete rank transform for improved logo recognition. In *IEEE International Conference on Communications*, pages 1–4, 2014. 1, 2

[2] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *IEEE International Conference on Computer Vision*, pages 1431–1439, 2015. 2, 7

[3] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):189–203, 2017. 5

[4] L. Dong, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang. Weakly supervised object localization with progressive domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2, 7

[5] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 4

[6] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 7

[7] 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. 1

[8] R. Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 2015. 1, 2

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[10] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. 3, 6

[11] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pages 3536–3544, 2014. 4

[12] S. C. Hoi, X. Wu, H. Liu, Y. Wu, H. Wang, H. Xue, and Q. Wu. Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks. *arXiv preprint arXiv:1511.02462*, 2015. 2, 3, 4

[13] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer. Deeplogo: Hitting logo recognition with the deep neural network hammer. *arXiv*, 2015. 2

[14] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *ACM International Conference on Multimedia*, pages 581–584, 2009. 1, 2, 3, 4

[15] Y. Kalantidis, L. G. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis. Scalable triangulation-based logo recognition. In *ACM International Conference on Multimedia Retrieval*, page 20, 2011. 1, 2

[16] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465, 2014. 1

[17] K.-W. Li, S.-Y. Chen, S. Su, D.-J. Duh, H. Zhang, and S. Li. Logo detection with extendibility and discrimination. *Multimedia tools and applications*, 72(2):1285–1310, 2014. 1, 2

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*. 2014. 4

[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, 2016. 2, 5, 7

[20] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the International Conference on Information and Knowledge Management*, 2000. 2, 5

[21] C. Pan, Z. Yan, X. Xu, M. Sun, J. Shao, and 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*, pages 123–126, 2013. 1

[22] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *British Machine Vision Conference*, volume 1, page 6, 2015. 3

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 5

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 1, 2, 5, 7

[25] J. Revaud, M. Douze, and C. Schmid. Correlation-based burstiness for logo retrieval. In *ACM International Conference on Multimedia*, pages 965–968, 2012. 1, 2

[26] S. Romberg and R. Lienhart. Bundle min-hashing for logo recognition. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 113–120. ACM, 2013. 1, 2

[27] S. Romberg, L. G. Pueyo, R. Lienhart, and R. Van Zwol. Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 25. ACM, 2011. 1, 2, 3, 4, 5

[28] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Seventh IEEE Workshop on Applications of Computer Vision*, 2005. 2

[29] 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):211–252, 2015. 1, 4, 5

[30] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for

recognition. In *Workshop of IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 7

[31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

[32] H. Su, X. Zhu, and S. Gong. Deep learning logo detection with data expansion by synthesising context. *IEEE Winter Conference on Applications of Computer Vision*, 2017. 2, 3, 4, 5, 6, 7

[33] X. Xu, T. Hospedales, and S. Gong. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, pages 1–25, 2017. 1

[34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014. 7

[35] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv*, 2015. 6

[36] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, 2014. 7