

# Scalable Logo Recognition using Proxies

István Fehérvári

istvanfe@amazon.com

Srikar Appalaraju

srikara@amazon.com

Amazon Inc.

## Abstract

Logo recognition is the task of identifying and classifying logos. Logo recognition is a challenging problem as there is no clear definition of a logo and there are huge variations of logos, brands and re-training to cover every variation is impractical. In this paper, we formulate logo recognition as a few-shot object detection problem. The two main components in our pipeline are universal logo detector and few-shot logo recognizer. The universal logo detector is a class-agnostic deep object detector network which tries to learn the characteristics of what makes a logo. It predicts bounding boxes on likely logo regions. These logo regions are then classified by logo recognizer using nearest neighbor search, trained by triplet loss using proxies. We also annotated a first of its kind product logo dataset containing 2000 logos from 295K images collected from Amazon called PL2K. Our pipeline achieves 97% recall with 0.6 mAP on PL2K test dataset and state-of-the-art 0.565 mAP on the publicly available FlickrLogos-32 test set without fine-tuning.

## 1. Introduction

Logo recognition has a long history in Computer vision with works dating back to 1993 [12]. While the problem is well defined (detect and identify brand logos in images), it is a challenging object recognition and classification problem as there is no clear definition of what constitutes a logo. A logo can be thought of as an artistic expression of a brand, it can be either a (stylized) letter or text, a graphical figure or any combination of these. Furthermore, some logos have a fixed set of colors with known fonts while others vary a lot in color and specialized unknown fonts. Additionally, due to the nature of a logo (as brand identity) there is no guarantee about its context or placement in an image, in reality logos could appear on any product, background or advertising surface. Also, this problem has large intra-class variations e.g. for a specific brand, there exist various logos types (old and new Adidas logos, small and big versions of Nike) and inter-class variations e.g. there exists logos which belong to different brands but look similar (see Figure 1).

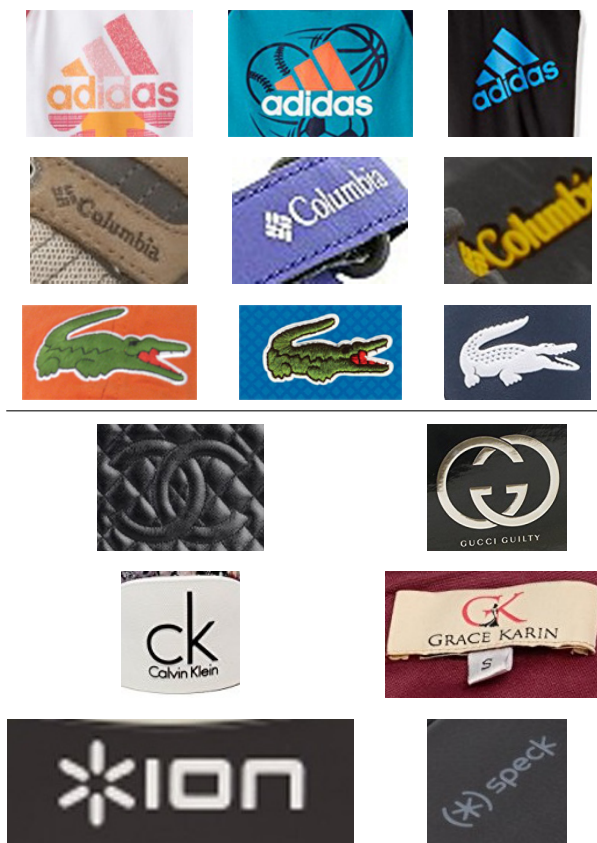


Figure 1. Logo variations exemplar images. (Row 1-3) Intra-class variations of brands Adidas, Columbia, Lacoste per row. Notice, different backgrounds, placement, fonts. (Row 4-6) Inter-class variations of brands Chanel - Gucci, Calvin Klein - Grace Karin, Ion - Speck. Notice, similar looking logos but belong to different brands.

Accurate logo recognition in images can have multiple applications. It can enable better semantic search, better personalized product recommendations, improved contextual ads, IP infringement detection amongst other applications.

Logo recognition has many inter- and intra-class variations, retraining with each new variation is unscalable. In

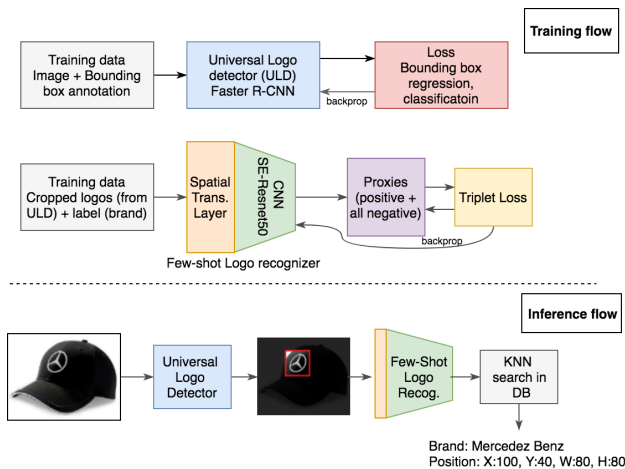


Figure 2. Flow diagram for our Architecture for training and inference. We train proxies jointly with our few-shot model which are used to compute the triplet loss.

this work, we explore logo recognition as a few-shot problem. We show experimentally and empirically that our approach is able to detect and identify previously unseen (in training) logos. We show that our models have better learned what makes something a logo than prior art, with performance being the evidence. We created a first of its kind Product logo dataset *PL2K*. It contains over 2000 logos with large inter- and intra-class variations. Our pipeline achieves 97% recall with 0.6 mAP on the new *PL2K* test dataset and state-of-the-art 0.565 mAP on the publicly available FlickrLogos-32 test set without fine-tuning. With this we present the main contributions of our work:

- Universal logo detector: a class-agnostic logo detector, capable of predicting bounding boxes on previously unseen logos.
- Novel logo recognizer: A network based on spatial transformer and proxy loss provides state-of-the-art results on FlickrLogos-32 test set. We further show by experiments that this type of architecture with metric learning does really well in few-shot logo recognition thereby providing a more generalized logo recognition model.
- Product logo dataset *PL2K*<sup>1</sup>: We discuss how we went about collecting and annotating this large-scale logo dataset from the Amazon catalog.

## 2. Related works

In this section we discuss closely related works in the fields of deep learning in computer vision, prior art in logo

<sup>1</sup>An image with a single product in front of a white background

recognition, and metric learning.

### 2.1. Deep learning

Our work adopts on recent deep learning research state-of-the-art results in image object detection and classification [15, 19, 25, 32, 37, 45, 46, 52, 58]. The problem of logo recognition itself has a rich research history. In 1990's the problem was mainly explored in information retrieval use-cases. An image descriptor was generated using affine transformations and stored in a database for retrieval [12]. There were also some neural network based approaches [7, 14] but the networks were not as deep nor the results as impressive as recent work.

In 2000's, with the advent of SIFT and related approaches [2, 38, 51] better image descriptors were possible. These methodologies were used to represent images better for logo recognition learning [6, 28, 44, 50, 70]. Apart from SIFT, other approaches were also explored mostly by the information retrieval community using min-hashing [49], metric learning [8], Vocab trees [41], using text [56], bundling features for improved search [67]. Most of these approaches needed complex image preprocessing pipelines along with several independent models.

Recent initiative in logo recognition use deep neural networks, which offer superior performance with end to end pipeline automation, i.e. from image and logo identification to recognition. Broadly speaking, the following approach is prevalent - an image is fed to deep neural object detector and classifier gives out predictions [4, 5, 21, 24, 43, 61, 62], most of these approaches use an ImageNet [32] trained CNN which is fine-tuned on FlickrLogos-32 [50] dataset. The lack of a big quality logo dataset makes the models less generalizable. Other datasets like WebLogo-2M [59] are large but this dataset is noisy with no manual bounding box annotations. Instead, the data is annotated via an unsupervised process, meaning the error rate is unknown. The Logos in the wild dataset is much better [62] but it lacks the large intra- and inter-class variations that *PL2K* provides.

### 2.2. Metric learning

Distance Metric learning (DML) has a very rich research history in information retrieval, machine learning, deep learning and recommender systems communities. DML has successfully been used in clustering [20], near duplicate detection [69], zero-shot learning [42], image retrieval [57]. We briefly cover it in relation to our work (deep methods). The seminal work in DML is to train a Siamese network with contrastive loss [9, 18], where pairwise distance is minimized for image pairs with same class label and push distance between dissimilar pairs greater than some fixed margin. A downside of this approach is that it focuses on absolute distances, whereas for most tasks, relative distances matter more. One improvement over contrastive loss is

triplet loss [55, 65] which constructs a set of triplets, where each triplet has an anchor, a positive, and a negative example where anchor and positive have the same class label and negative has a different class label.

In practice, the performance of these methods heavily depends on pair mining strategies as there are exponential number of pairs (mostly negative pairs) that can be generated. Several works in recent years have explored various smart pair mining strategies. Facenet [54] proposed online hard negative mining strategy but this technique has a shortcoming where it empirically required larger batch sizes to work. Curriculum learning [3] was explored by [1] where a probability distribution was used to sample image pairs online in order of their hardness, with easy to cluster image pairs sampled more earlier on in training and harder to identify pairs introduced later on in training. There exists other DML sampling approaches trying to devise losses easier to minimize using small mini-batches [20, 26, 42, 57, 64, 66].

Relating this prior art with our contributions in logo research; none of these approaches explore logo recognition as a few-shot clustering formulation which we feel is a better fit for this problem. As a result, we did not have to perform class imbalance correction (as done by [59]), our approach can handle large number of logo classes and do effective few-shot logo detection by projecting new logos into an embedding space. We used a combination of triplet-loss and proxies[40] to optimize this embedding space and not a simple distance measure. We hypothesize that the principle of proposed method can be applied effortlessly to other image tasks like classification or object detection. We picked logo recognition due to its wide range of applications and deep research history.

### 3. Approach

As discussed earlier, one of the key challenges with logo detection is that the context in which the logo is embedded can vary almost infinitely. State of the art deep learning object detectors that are trained to localize and identify a closed set of logos will inherently use the contexts of each logo for training and prediction which makes them susceptible to context changes. For example, a logo that appears only on shoes in the training data might remain invisible or get confused by same logo if it is displayed on a coffee mug during inference.

To overcome this issue, we propose a two-step approach, where first a semantic logo detector identifies rectangular regions of an image where a logo might be located and a second model, logo recognizer identifies its class/brand. In contrast to recent works [62] that apply state of the art object detectors such as Faster R-CNN [48] or SSD[37] to detect and identify a fixed set of logos, we aim at a universal logo detector that does not need further retraining if the classes change. This method also alleviates the problem of collect-

ing and annotating a new large body of training data for every future logo that needs to be detected.

Given a large number of training images across a wide range of brands and contexts, we expect the models to learn the abstract concept of *logoness* and to be able to work with any logo class at inference time. In practice, we train these models in a class-agnostic way: every generated region proposal is classified in a binary fashion: logo or background discarding any class-related information. In Section 5 we discuss results of this claim of universal logo detection on *PL2K* dataset and on public logo dataset FlickrLogos-32. We also run different state-of-the-art object detector architectures (SSD, Faster R-CNN, YOLOv3) and analyze their results.

#### 3.1. Few-shot Logo Recognition

Once the semantic logo detector has identified a set of probably logo regions within a given image we need to have a mechanism that can correctly classify these regions into its corresponding logo class/brand. Ideally, this step could be solved via a state of the art CNN image classification model such as ResNets[19] with multi-class classification. However this necessitates the right amount of training data for every class, class imbalance corrections and might also constrain the number of classes.

Recent advances in deep embedding learning propelled the research in few- [30, 68] or one- [53] or zero-shot learning [33] where the aim is to use only a few, single or no examples of each class during training. The typical way this is achieved is via metric learning, where a model learns the similarity among arbitrary groups of data, thus being able to cope even with a large number of (unseen) classes.

Currently, state-of-the-art methods for metric learning employ deep (convolution) neural networks, which are trained to output an embedding vector for each input image so that it minimizes a loss function defined over the distances of points. Usually, distances are learned using triplets of similar and dissimilar points  $D = (x, y, z)$ , where  $x$  being the anchor,  $y$  the positive, and  $z$  the negative point and  $d$  is the Euclidean distance function. With  $y$  being more similar to  $x$  than  $z$  the task is to learn a distance respecting the similarity relationships encoded in  $D$ :

$$d(x, y) \leq d(x, z) \text{ for all } (x, y, z) \in D \quad (1)$$

Triplet-loss addresses this with a hinge function to create a fixed margin between the anchor-positive difference, and the anchor-negative difference:

$$L_{triplet}(x, y, z) = [d(x, y) + M - d(x, z)]_+ \quad (2)$$

However, it has been shown that the performance of these functions depends greatly on the way these pairs and triplets are sampled [1, 54]. In fact, computing the right set of

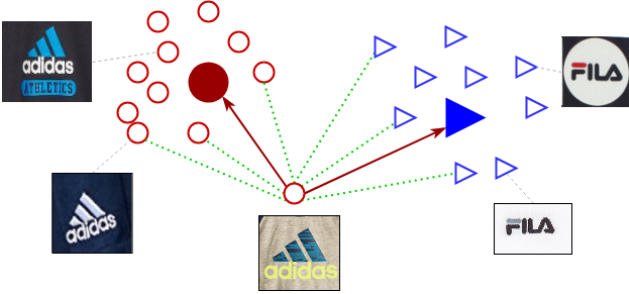


Figure 3. Example of proxies used during training: for the Adidas logo (in the middle) distances are computed to the positive (dark red) proxy to its left and negative (blue) proxy to its right instead of other training images (small circles / triangles). Proxies do not belong to any single image, they are learned during training.

triplets is a computationally expensive task which has to be performed for every mini-batch during training for optimal results. Movshovitz-Attias *et al.* introduced the notion of proxies [40] in combination with *NCA* loss [17] that completely removes the sampling burden while providing state-of-the-art performance on CUB200 [63], Cars196 [31] and Stanford Products[42] datasets. They [40] define *NCA* loss over proxies the following way:

$$L_{NCA}(x, y, Z) = -\log \left( \frac{\exp(-d(x, p(y)))}{\sum_{z \in Z} \exp(-d(x, p(z)))} \right), \quad (3)$$

where  $Z$  is a set of all negative points for  $x$  and  $p(x)$  is the *proxy* for  $x$  with  $p(x) = \arg \min d(x, p)$  for all  $p \in P$  that we need to learn. See Figure 3 for an illustrative example of proxies. Similar to the original work we train our model and all proxies with the same norm:  $N_p = N_x$ . Since *NCA* loss even with proxies over-fit very early for our logo identification problem, we used the original triplet loss with proxies:

$$L_{triplet}(x, y, Z) = [d(x, p(y)) + M - d(x, p(Z))]_+, \quad (4)$$

We choose one proxy per logo class with static proxy assignment. This yields fast convergence and state-of-the-art results on FlickrLogos-32 without using any triplet-sampling strategy. In the experiments section we show how this simple change in the loss function leads to superior performance without sampling.

At inference time, using this trained embedding function, one can apply (approximate) K-nearest neighbor search among embedding vectors to find the corresponding prediction for each query image.

#### 4. Product Logo Dataset (PL2K)

Ideally, our training set would be a large body of annotated images featuring logos across a high variety of contexts and domains to be able to train a deep-learning based

class-agnostic object detector model as these models have millions of trainable parameters. Unfortunately, no current public dataset satisfies these requirements. One reason could be that image collection and annotation for such a dataset would require expensive manual work to keep quality high and to prevent potential copyright issues that might arise when images are farmed with automated scripts from the Internet. Also care has to be taken to not annotate counterfeit logos as we would not want the models to learn the wrong logo representation.

Therefore, we decided to build a new dataset by sampling images from the Amazon Product Catalog for the following reasons:

- A large body of publicly accessible images are available, thus easy to automate data collection.
- Images are labeled with the corresponding brand that helps with annotation.
- We are interested in the abstraction capacity of object detection models i.e. how they perform on non-product images.

Product images on Amazon typically feature a single or multiple products in front of a white background. Our working hypothesis is that a large amount of product images will offer a high enough variance in logo contexts from which the object detection model can learn and generalize the concept of a Logo. Furthermore, we chose 2000 brands based on popularity that satisfy the following conditions: (i) have a significant number of images (ii) well-established logo (iii) logo frequently used on the product.

We sampled a total of 1 million product images randomly from these brands and used Amazon Mechanical Turk to annotate them. Every image was sent to 9 different *MTruk* workers for annotation. Each worker had to complete the following task:

1. Identify if the image contains no, one or multi logos; label image as such *NO LOGO*, *ONE LOGO*, *MULTIPLE LOGO*.
2. For the bounding box: in a separate task, of the images with at least one logo, the workers were instructed to locate the leftmost, topmost logo on the image (if multiple present) and draw a rectangle around it. If there were still multiple options, the workers were instructed to choose the biggest one.

**Post-processing:** Due to the different interpretations of the term logo, we received very different annotations for the same image from multiple workers. To consolidate the results, we filtered out every image marked as *NO LOGO* by more than 3 workers (out of 9). This finally gave us 295,814 images, each with at least 6 bounding-box annotations.

Dataset	Logos	Images	Supervision	Noisy	Construction	Scalability
BelgaLogos [29]	37	10,000	Object-Level	✗	Manually	Weak
FlickrLogos-32 [50]	32	8,240	Object-Level	✗	Manually	Weak
FlickrLogos-47 [50]	47	8,240	Object-Level	✗	Manually	Weak
TopLogo-10 [60]	10	700	Object-Level	✗	Manually	Weak
Logo-NET [22]	160	73,414	Object-Level	✗	Manually	Weak
WebLogo-2M [59]	194	1,867,177	Image-Level	✓	Automatically	Strong
Logos in the wild [62]	871	11,054	Object-Level	✓	Manually	Medium
<b>PL2K (Ours)</b>	2000	295,814	Object-Level	✗	Semi-automatically	Strong

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

Image set	No. of images	No. of brands
Training	185247	206
Validation	46312	
Testing	57970	1528
Negatives	10000	2000

Table 2. PL2K data split for train and test.

In order to reduce noise and accurately merge the annotation box we used the DBScan clustering algorithm [13] as it requires no *a priori* knowledge on the number of clusters. The algorithm was performed on a precomputed pairwise distance matrix of the annotation rectangles where the distance was defined as the complement of intersection over union (IoU), see equation 5. We empirically derived epsilon to be 0.6 and the minimum core samples to be 1 as these yielded the best results.

$$D = 1 - \frac{R_1 \cap R_2}{R_1 \cup R_2}. \quad (5)$$

As we started using *PL2K*, we found that several annotators marked the whole image as a logo even though that was clearly incorrect. Therefore, we removed all merged bounding boxes that have an  $IoU > 0.65$  with the image rectangle. We removed nearly 36k bounding boxes and about 2K images from the dataset. Finally, we split the dataset into training plus validation (80%) and testing (20%) making sure that sets do not share images and brands taken from the same set. Ultimately, we are interested in the generalization capability of the model on unseen brands.

We also added a negative set that consists of randomly sampled images that were marked as containing no logos by all annotators. We use this set to measure the false positive rates of the models. Table 2 provides a quick overview of the *PL2K* dataset while Table 1 compares *PL2K* with existing logo detection datasets. Note, that we split the data in such a way that there are almost 8 times more logos in test data than in train data. We wanted to showcase the universal object detector’s generalization capabilities.

**Data for Few-shot logo detector:** The second part of our data collection effort was for few-shot logo detector.

This was slightly different from universal logo detector. This model operates on logo regions i.e. logo appearances cropped out from images. We picked product images for the top 242 logos and ran them through our universal logo detector. Based on the regions proposed by universal logo detector, we manually filtered out false positive regions and identified 242 valid brand logos. From this we had at least 700 cropped regions for each logo. This dataset was then split into a train and test set (80/20%) with 193 and 49 logo classes respectively.

## 5. Experiments

We split our experiments into two parts: first we investigate the performance of the Universal logo detector operating on *PL2K* and FlickrLogos-32. As discussed in table 1, *PL2K* is relatively clean Amazon catalog images and FlickrLogos-32 are real-world logos in the wild dataset. Second, we discuss our experiments with the few-shot logo classifier which works with cropped image regions. Finally in end-to-end section, we see how these techniques perform on FlickrLogos-32 without fine-tuning on this dataset.

### 5.1. Universal Logo Detection

The following three state-of-the-art object detector architectures with two output classes were tested for our universal logo detector: Faster R-CNN [48], SSD [37], and YOLOv3 [47]. Faster R-CNN is a two step detector that has a higher performance on standard datasets (e.g. MS COCO[36]), but is a lot slower than the other two single-shot detectors. There have been several proposals on improving the performance of this model [34, 35] but these only offer a minor increase in mAP at the cost of speed.

We used a ResNet50 [19] base CNN for all networks except for *YOLOv3* which uses the Darknet53 architecture. All of the networks were pre-trained on ImageNet [10] and then trained end-to-end on the *PL2K* dataset with a fixed input size of 512x512 for 20 epochs. YOLOv3 was trained with randomly resized inputs in the range of [320, 640] with steps of 32, to achieve better accuracy. We computed the recall at  $IoU > 0.5$ , average precision (AP), and the number of regions generated on the negative/no logo set.

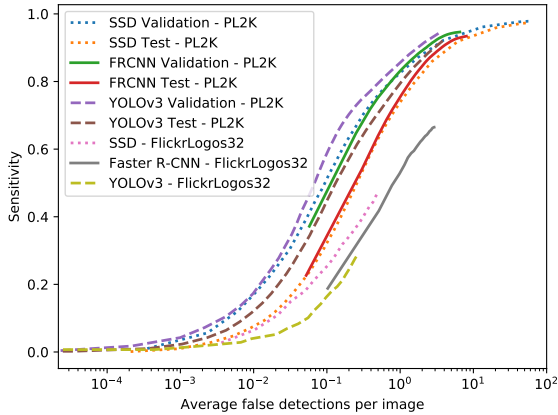


Figure 4. FROC curves of the three detector models on FlickrLogos-32. Faster R-CNN achieves higher recall but with lot more false positives. Best viewed in color.

**Same domain (PL2K).** We find that all models achieve very high recall and AP values with SSD having the highest recall and YOLOv3 the highest AP on the *PL2K* validation dataset. This trend repeats on the test set with a 0.1 drop in AP and 1-2% points in recall meaning the models still perform well on a wide range of unseen logos and products. Interestingly, the two-step detection by Faster R-CNN does not bring any advantage in performance. On the contrary, it increases the false positive rate by a large margin. The FROC curves on Figure 4 depict this behavior accurately.

**Different domain (FlickrLogos-32).** In order to see how these models work on a completely different domain without fine-tuning we ran the same evaluation on the FlickrLogos-32 dataset. This dataset is the most popular evaluation dataset for logos, consisting of 8,240 images covering 32 logos/brands. The performance trend seemed to reverse: Faster R-CNN with an accuracy of close to 80% and AP of 0.42 outperforms SSD and YOLOv3 by a large margin. This suggests that Faster R-CNN learned less domain-specific features which combined with the almost 8x more predictions outperforms all other open-set detectors reported by [62]. See Table 3 for the full set of results and Figure 4 for the corresponding FROC curves.

Based on these experiments, we chose Faster R-CNN as it has superior generalization capabilities. This model is what helped achieve the state-of-the-art results on FlickrLogos-32 (see table 5). More analysis revealed that SSD and YOLOv3 had issues detecting smaller bounding box regions, images with occlusions or logo which blend into their environment.

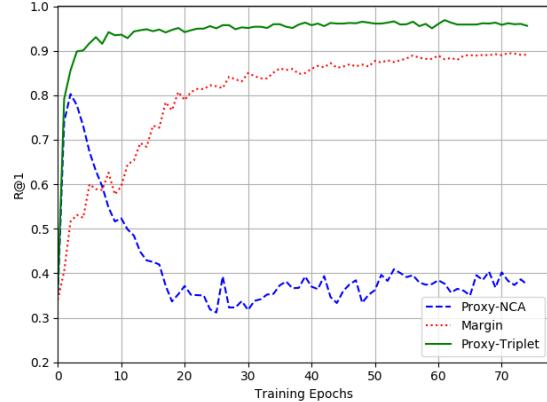


Figure 5. Performance of various distance learning loss functions for Few-shot model on the *PL2K* test set.

## 5.2. Few-shot Logo Identification

For the few-shot logo embedding model we used the SE-Resnet50 [23] architecture with the same modifications as described in[11]. Input images were resized to 160x160 pixels, the embedding dimension was 128 and the batch size 32. We used the Adam optimizer with momentum 0.9, weight decay 0.0005, and learning rate  $10^{-4}$  which we reduced by a factor of 0.8 every 20 epochs. The network’s parameters were initialized using Xavier initialization [16] with magnitude 2, no transfer learning was used.

We trained few-shot model by passing *PL2K* annotated images to the few-shot logo detector (section on *PL2K* dataset). The 242 logo classes were used for training and testing without any sampling strategy. Various loss functions and spatial transformer layers (see table 4) were tried on top of this base architecture.

For comparison, we trained the same model using distance-weighted sampling with margin-based loss[66] as well as proxy-NCA loss since both methods report higher performance than triplet loss with various sampling strategies. As a solid baseline we also added cross-entropy loss, though here we are using the last layer of the feature extractor, thus the embedding dimension is much larger than 128.

As seen on Figure 5 the proxy-triplet loss converges very fast to a superior score in contrast with the other approaches. For the first few epochs proxy-NCA has almost identical performance then it starts to diverge and quickly decline (over-fit). This suggests that proxy-augmented loss functions are strongly dataset and/or hyper-parameter sensitive but it needs further investigation.

Most logos do include words and other characteristics which could provide some hint about the right orientation of a logo in noisy contexts, we also experimented with adding a Spatial Transformer network (STN) layer [27] to

Model	Validation set		Test set		Negative set	FlickrLogos-32		
	Recall	AP	Recall	AP	No. detections	Recall	AP	Negative detections
Faster R-CNN	94.76%	0.72	93.52%	0.63	147152	<b>79.87%</b>	<b>0.42</b>	8379
SSD	<b>98.05%</b>	0.73	<b>97.73%</b>	0.62	<b>19295</b>	60.04%	0.38	1136
YOLOv3	94.29%	<b>0.77</b>	92.10%	<b>0.70</b>	19504	44.69%	0.22	<b>985</b>

Table 3. Universal Logo Detector performance on the *PL2K* and FlickrLogos-32 datasets.

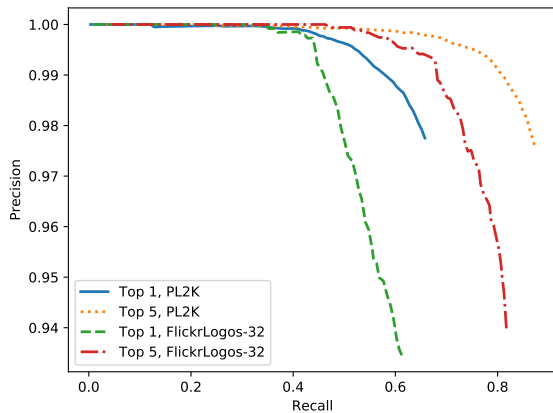


Figure 6. Precision-Recall metric for few shot logo detector on *PL2K* and FlickrLogos-32 data. Note the state-of-the-art performance of the few-shot logo detector on Flickr-32 data. The model has not seen FlickrLogos-32 data in training. Best viewed in color.



Figure 7. Retrieval results of the few-shot model on the FlickrLogos-32 and *PL2K* datasets. Left column contains query regions.

Few-shot Logo model	Top 1 Recall
CrossEntropy Loss	89.10%
Margin Loss	91.09%
Proxy-NCA Loss	80.26%
Proxy-Triplet Loss	96.80%
Proxy-Triplet Loss with STN	<b>97.16%</b>

Table 4. Top1 recall of few-shot Resnet50 model and various losses on *PL2K* dataset.

the proxy-triplet model, which improved the accuracy by a small margin. See the table 4 below for the best top 1 recall scores. Fig. 6 shows the performance of the final few-shot model on *PL2K* and FlickrLogos-32 dataset.

### 5.3. Qualitative Analysis

To get a better understanding on quality of the trained solution we ran the t-SNE algorithm [39] on a randomly selected subset of test classes for 1000 iterations with perplexity 40 (see Figure 8). We find that our model successfully separates very similar looking logo classes even if they use similar font or color. There are a few single points scattered around the space (in the middle) these are impressions from logos that are radically different in the use of color, font than the rest of the logos.

Figure 7 illustrates some of the successful retrieval results on both datasets. In Figure 9, we show a few examples that are wrongly classified (right column) by our model based on query image (left column). Diving deeper into the mis-classified examples suggested low resolution images, logo consisting of very simple shapes being a few reasons for mis-classification.

### 5.4. End-to-end evaluation

We evaluate the performance of our universal detector combined with the few-shot identification model on FlickrLogos-32 a popular logo datasets and compare it to the state-of-the-art. None of the models were trained or fine-tuned on FlickrLogos-32 dataset. All models were trained on *PL2K* dataset. As seen in Table 5 and table 4, the final model used was Faster R-CNN as universal Logo detector; few-shot model was a SE-Resnet50 with proxy-triplet loss with Spatial transformer layers. Top 5 accuracy worked best as it generated more proposals.

For few-shot we extracted the first five ground truth re-

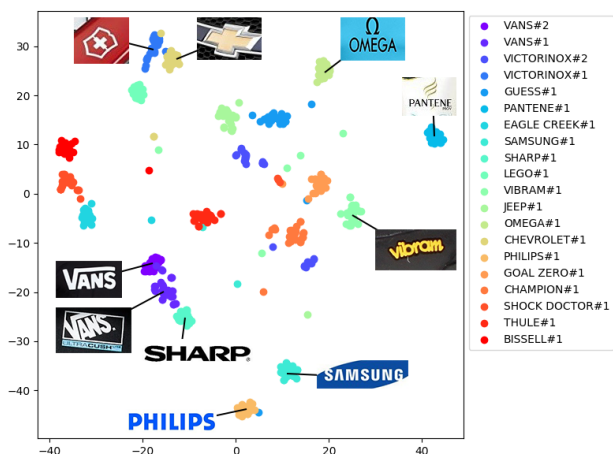


Figure 8. t-SNE [39] plot of a random subset of test classes. The model successfully maps logos of the same class close to each other, even with high inter- and intra-class variations (e.g. VANS#1-#2, or Samsung and Philips). Best viewed in color.



Figure 9. Erroneous classifications: few-shot recognition becomes hard when image resolution is low and/or logos are made of very simple shapes

gions per brand and used it as anchors. These anchor regions were excluded from evaluation to avoid bias. We also ran two evaluations per each detector type (single and five shot). Faster R-CNN worked best with mAP of 0.56558. mAP is decided by region proposals with class detection threshold of 0.5.

## 6. Conclusion and Future work

In this work we shared our approach to few shot logo recognition using deep learning two stage models. We trained our models on *PL2K* and evaluated on FlickrLogos-32 to achieve new state-of-the-art performance of 56.55% mAP@5. This empirically indicates that our approach does

End to End Model	mAP@1	mAP@5	No. proposals
SSD + FS*	35.833%	44.79%	2655
Faster R-CNN + FS*	<b>44.42%</b>	<b>56.55%</b>	8786
YOLOv3 + FS*	23.31%	30.12%	1525
Tüzkö et al. [62]	46.4%		N/A

Table 5. Top 1 and Top5 Accuracy of the end to end evaluation on FlickrLogos-32 dataset. This uses both Universal Logo detector and Few-shot Logo recognizer. \*FS=Few-shot model SE-Resnet50 with Proxy-Triplet Loss with STN as shown in table 4 (Row 1-3). Row 4 shows the results of the only reported open-set detector. Note, that this system was trained using FlickrLogos-32.

good domain adaptation without fine-tuning.

We also conducted extensive experiments on various CNN architectures and compared with existing logo works to show that triplet-loss with proxies is an effective way to find similar images. We also presented product logo dataset *PL2K*, a first of its kind large scale logo dataset.

Future extensions of this works could look at application of this work in broader contexts of image similarity search, generic object detection or going deeper into understanding why triplet-loss with proxies does so well and its limitations.

## 7. Acknowledgments

We would like to thank Arun Reddy and Mingwei Shen for giving us the opportunity to work on this problem. We would also like to thank Niel Lawrence, Rajeev Rastogi, Avi Saxena, Ralf Herbrich, Fedor Zhdanov, Dominique L'Eplattenier, Marc Ascolese and Ives Macedo for providing constructive feedback on our work. Finally, Ankit Raizada for his work on putting together the *PL2K* dataset and John Weresh for naming it.

## References

- [1] S. Appalaraju and V. Chaoji. Image similarity using Deep CNN and Curriculum Learning. *arXiv preprint arXiv:1709.08761*, 2017.
- [2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.
- [3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.
- [4] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini. Logo recognition using CNN features. In *International Conference on Image Analysis and Processing*, pages 438–448. Springer, 2015.

- [5] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini. Deep learning for logo recognition. *Neurocomputing*, 245:23–30, 2017.
- [6] R. Boia, A. Bandrabur, and C. Florea. Local description using multi-scale complete rank transform for improved logo recognition. In *Communications (COMM), 2014 10th International Conference on*, pages 1–4. IEEE, 2014.
- [7] F. Cesarini, E. F. M. Gori, S. Marinai, J. Sheng, and G. Soda. A neural-based architecture for spot-noisy logo recognition. In *icdar*, page 175. IEEE, 1997.
- [8] J. Chen, M. K. Leung, and Y. Gao. Noisy logo recognition using line segment hausdorff distance. *Pattern recognition*, 36(4):943–955, 2003.
- [9] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.
- [11] J. Deng, J. Guo, and S. Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CoRR*, abs/1801.07698, 2018.
- [12] D. S. Doermann, E. Rivlin, and I. Weiss. Logo recognition using geometric invariants. In *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pages 894–897. IEEE, 1993.
- [13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.
- [14] E. Francesconi, P. Frasconi, M. Gori, S. Marinai, J. Sheng, G. Soda, and A. Sperduti. Logo recognition by recursive neural networks. In *International Workshop on Graphics Recognition*, pages 104–117. Springer, 1997.
- [15] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [16] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.
- [17] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov. Neighbourhood components analysis. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 513–520. MIT Press, 2005.
- [18] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, June 2006.
- [19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [20] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 31–35. IEEE, 2016.
- [21] 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.
- [22] S. C. H. 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. *CoRR*, abs/1511.02462, 2015.
- [23] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 7, 2017.
- [24] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer. Deeplogo: Hitting logo recognition with the deep neural network hammer. *arXiv preprint arXiv:1510.02131*, 2015.
- [25] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [26] C. Ionescu, O. Vantzos, and C. Sminchisescu. Training deep networks with structured layers by matrix backpropagation. *arXiv preprint arXiv:1509.07838*, 2015.
- [27] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *CoRR*, abs/1506.02025, 2015.
- [28] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 581–584. ACM, 2009.
- [29] A. Joly and O. Buisson. Logo retrieval with a contrario visual query expansion. In *Proceedings of the 17th ACM International Conference on Multimedia, MM '09*, pages 581–584, New York, NY, USA, 2009. ACM.
- [30] A. Kimura, Z. Ghahramani, K. Takeuchi, T. Iwata, and N. Ueda. Few-shot learning of neural networks from scratch by pseudo example optimization. *BMVC*, 2018.
- [31] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [33] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015.
- [34] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.
- [35] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017.

- [36] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.
- [37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [38] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [39] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.
- [40] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *ICCV*, pages 360–368. IEEE Computer Society, 2017.
- [41] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. Ieee, 2006.
- [42] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016.
- [43] G. Oliveira, X. Frazão, A. Pimentel, and B. Ribeiro. Automatic graphic logo detection via fast region-based convolutional networks. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 985–991. IEEE, 2016.
- [44] A. P. Psyllos, C.-N. E. Anagnostopoulos, and E. Kayafas. Vehicle logo recognition using a sift-based enhanced matching scheme. *IEEE transactions on intelligent transportation systems*, 11(2):322–328, 2010.
- [45] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [46] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [47] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.
- [48] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, June 2017.
- [49] 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.
- [50] 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, ICMR '11*, pages 25:1–25:8, New York, NY, USA, 2011. ACM.
- [51] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.
- [52] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.
- [53] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. One-shot learning with memory-augmented neural networks. *arXiv preprint arXiv:1605.06065*, 2016.
- [54] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.
- [55] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in neural information processing systems*, pages 41–48, 2004.
- [56] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.
- [57] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016.
- [58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [59] H. Su, S. Gong, X. Zhu, et al. Weblogo-2m: Scalable logo detection by deep learning from the web. 2018.
- [60] H. Su, X. Zhu, and S. Gong. Deep learning logo detection with data expansion by synthesising context. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 530–539, 2017.
- [61] H. Su, X. Zhu, and S. Gong. Open logo detection challenge. *arXiv preprint arXiv:1807.01964*, 2018.
- [62] A. Tüzkö, C. Herrmann, D. Manger, and J. Beyerer. Open set logo detection and retrieval. *arXiv preprint arXiv:1710.10891*, 2017.
- [63] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [64] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.
- [65] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.
- [66] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krähenbühl. Sampling matters in deep embedding learning. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.
- [67] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, pages 25–32. IEEE, 2009.
- [68] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 2018.
- [69] S. Zheng, Y. Song, T. Leung, and I. Goodfellow. Improving the robustness of deep neural networks via stability training.

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4480–4488, 2016.

- [70] G. Zhu and D. Doermann. Automatic document logo detection. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 864–868. IEEE, 2007.