Image-Text Pre-Training for Logo Recognition

Mark Hubenthal

mhubenth@amazon.com

Amazon Inc.

Suren Kumar

ssurkum@amazon.com

Amazon Inc.

Abstract

Open-set logo recognition is commonly solved by first detecting possible logo regions and then matching the detected parts against an ever-evolving dataset of cropped logo images. The matching model, a metric learning problem, is especially challenging for logo recognition due to the mixture of text and symbols in logos. We propose two novel contributions to improve the matching model's performance: (a) using image-text paired samples for pre-training, and (b) an improved metric learning loss function. A standard paradigm of fine-tuning ImageNet pre-trained models fails to discover the text sensitivity necessary to solve the matching problem effectively. This work demonstrates the importance of pre-training on image-text pairs, which significantly improves the performance of a visual embedder trained for the logo retrieval task, especially for more text-dominant classes. We construct a composite public logo dataset combining LogoDet3K, OpenLogo, and FlickrLogos-47 deemed OpenLogoDet3K47. We show that the same vision backbone pre-trained on image-text data, when fine-tuned on OpenLogoDet3K47, achieves 98.6% recall@1, significantly improving performance over pre-training on Imagenet1K (97.6%). We generalize the ProxyNCA++ loss function to propose ProxyNCAHN++ which incorporates class-specific hard negative images. The proposed method sets new state-of-the-art on five public logo datasets considered, with a 3.5% zero-shot recall@1 improvement on LogoDet3K test, 4% on OpenLogo, 6.5% on FlickrLogos-47, 6.2% on Logos In The Wild, and 0.6% on BelgaLogo.

1. Introduction

Logo Recognition, the problem of identifying both logo regions and their associated logo classes, has several essential application areas, including brand recognition, contextual advertising, and trademark infringement detection [36]. Research in this field has closely tracked advances in general image recognition, given the similarities between the two problems. The field has moved from hand-designed ge-ometric invariants [7] to Scale-Invariant Feature Transforms (SIFT) [24] and is now driven entirely by large deep neural networks [2]. Two unique factors separate logo recognition from the deep learning-based image recognition trajectory. First, unlike typical object detection or image classification problems, new logo classes are constantly emerging as new companies are registered or contemporary styles are adopted. Second, an archetypal logo class can have an abstract definition with high intra-class variation, including different colors, styles, texts, and image backgrounds. Several research teams have addressed the first challenge by extracting a logo-agnostic bounding box and then leveraging metric learning to match detected regions against a potentially evolving set of logo images. However, metric learning approaches have struggled to perform robustly on the logo region matching problem due to high intra-class variations and the text-heavy nature of logo regions [21].

We hypothesize that the lack of text sensitivity in deep neural networks for metric learning is due to constraints in the fine-tuning process. The success of deep learning models on computer vision problems is driven by pre-training on large open-source datasets such as ImageNet and then adapting network weights to a downstream task. The performance of the resulting networks is heavily dependent on the characteristics of the pre-training dataset and the task similarity between fine-tuning and pre-training stages. However, popular pre-training datasets such as ImageNet or MS-COCO have limited text present in the images they contain. Thus, the current fine-tuning paradigm has severely limited performance on tasks that require text understanding, such as logo recognition and text-based visual question answering. Researchers have attempted to overcome such challenges around text representation by explicitly extracting optical character recognition (OCR) information and representing it for image captioning and visual question answering [40, 17, 42]. Text-aware embeddings are crucial for open-set logo recognition because logos often contain text content. Li et al. [21] proposed explicitly leveraging the brand name as a target to build a parallel text-in-image representation to bridge the gap between the representative capability of traditional vision-backbones and the

demands of logo recognition. Hu. et al. [13] proposed fusing explicit OCR with a textual description of an image in a multimodal fusion framework. However, such approaches increase overall embedding size, require additional downstream supervision data, and achieve limited performance gains.

We propose building on the recent success of large multimodal pre-training to improve text sensitivity of metric learning network backbones. Our solution, similar to existing works such as CLIP [27] and ALIGN [14], leverages normalized softmax-based constrastive learning to pre-train vision networks that generate text-sensitive embeddings. When adapted to the logo recognition problem, such networks significantly improve performance without requiring additional supervision, a separate text processing backbone, or OCR extraction. We also improve the fine-tuning process of the metric learning backbone with a new, improved loss function, deemed ProxyNCAHN++, that can leverage hard negatives in or out of domain. This helps to address the issue of very similar logos or the potential similarity between a logo and some out-of-domain designs or patterns.

In summary, the key contributions of our work are

1. We propose using image-text multimodal pre-training to improve text sensitivity of metric learning backbones for open-set logo recognition.

2. We present a new loss function to increase inter-class distance despite significant intra-class variation. An ablation study is documented in Section 4.4.

3. We perform a rigorous evaluation on several open-source logo recognition datasets in Section 4.3, where our solution surpasses state-of-the-art.

2. Related Work

2.1. Logo Recognition

The problem of logo recognition has closely tracked the progress in the broader image recognition field. In one of the early works of 1993, Doermann et al. [7] designed many geometrical invariants such as image moments to match the features of a new logo image against a database. The transition to better hand-crafted features such as Scale-Invariant Feature Transforms (SIFT) [24] were readily adopted by the logo recognition community [15]. Early attempts in the late 90s at leveraging neural networks [4] for logo recognition problems achieved limited success. That has changed in the past few years due to better neural network architectures [12], training techniques [20] and availability of large datasets [5] and GPUs. Bianco et al. [2] proposed building on the success of the Fast Region-Based Convolutional Network [10] recognition pipeline by first sampling image regions and then classifying them into a logo class.

Su et. al. [34] utilized improvements in the object detection literature, specifically Faster R-CNN [29] and YOLO [28]. They further relaxed the requirement of localized logo bounding boxes by synthetically compositing logo images onto background images. Wang et al. [39] introduced a large dataset of 3K logo categories and made improvements to the YOLO framework. These methods made substantial gains in logo-recognition performance but were only effective in the closed-set setting. Adding any new logo at test time would require re-training such models.

Tuzko et al. [37] proposed one of the first deep-learning-based open-set logo recognition methods. They use a Faster R-CNN [29] network to detect any logo region in the image and use an ImageNet-pre-trained network to match the logo region against a dataset of cropped logos. Fehérvári et al. [9] proposed taking advantage of metric learning approaches to improve the matching process further. This two-step paradigm of class-agnostic logo region localization and subsequent feature matching against an ever-evolving set of logos has become the norm [1]. The key challenge is to ensure the robustness of the metric learning backbone, specifically its text sensitivity. Typical text-invariance is due to initialization from ImageNet pre-trained networks and a lack of direct OCR supervision. Some fixes have been proposed to close this gap. Li et al. [21] proposed a separate branch of text feature extraction to generate the text contained in the logo. Hu. et al. [13] explicitly extract OCR information from the image and use an associated image description process through BERT [6] to extract brand information in the image. In contrast, we propose a simple change to the matching network's pre-training strategy, which does not require additional training data or increase in computational cost. Our method uses image-text multimodal pre-training, which imbues the vision network backbone with OCR capabilities.

2.2. Image-Text Pretraining

The use of attention-based methods in computer vision such as Vision Transformers [8] has led to many works that jointly model image and text. Vision and language transformer (ViLT) [19] proposed concatenating words with image patches to obtain a joint representation. ViLT achieved impressive results on multiple downstream applications, including visual question answering and image captioning. Li et al. [22] demonstrated multimodal training on a noisy set of image-text correspondences without the need for detailed annotations. We are particularly enthused by Contrastive Language-Image Pre-Training (CLIP) [27] where the authors proposed a simple yet effective method for multimodal training. CLIP constructs a large image-text dataset, WebImageText (WIT), by extracting images and their corresponding descriptions (such as alt text) and uses the resulting pairs to contrastively train image and text encoder
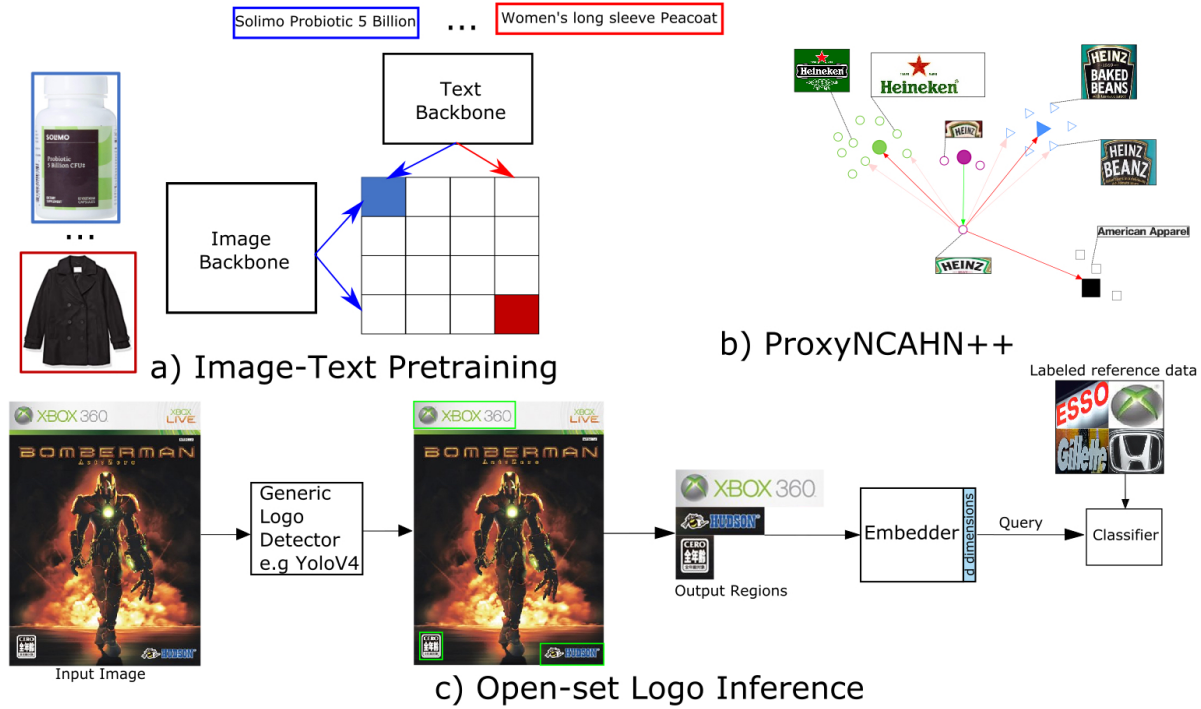
Figure 1. Overall setup of the proposed open-set logo recognition pipeline. a) Image-Text Pretraining: A batch of images with corresponding text titles is independently passed through a vision and a text backbone to extract embeddings. Both the backbones are trained by maximizing the alignment between the corresponding pairs ensuring a diagonally dominant dot product similarity matrix. b) The pretrained image backbone is further fine-tuned to serve as an embedder for the logo recognition problem using a new metric loss function, ProxyNCAHN++. Filled-in shapes represent learned proxy vectors and unfilled shaped represent image embedding vectors. For an input image from class h_j_heinz, it is attracted to its own class' proxy (green arrow), pushed away from other class proxies (red arrows), and also pushed away from individual image embeddings belonging to assigned hard negative classes heinz_baked_beans and heineken_text (pink arrows). c) Inference Pipeline: A generic logo detector extract all possible logo bounding boxes during inference. We then extract embeddings of these individual boxes (with the embedder trained in Step (b)) and compare against a dataset of evolving logos to compute the class labels.

backbones. The image backbone thus trained beats all other pre-trained methods on datasets that require an understanding of the text in the image, such as HatefulMemes [18].

In addition to experimenting with OpenAI's CLIP-trained vision transformer (ViT) model, we train an image-text backbone on a large dataset of product image-title pairs from an e-commerce website for comparison. The trained network demonstrates significant improvement on logos that contain text, as shown by Figure 3. Next, we discuss the loss function used to fine-tune the matching network's weights.

2.3. Metric Learning Loss Function

The three key attributes of any embedder/matcher are the network backbone and its pre-training, the loss function, and the metric used to compare the extracted embeddings. Training a matching model aims to push the embeddings of related logo images closer while moving apart image embeddings of different logos. Some early works

such as pair contrastive loss [11] were adopted in designing losses like the triplet loss [31] to better structure the embedding space. However, a common difficulty in pair-wise, triplet, or higher-pair loss functions has been the issue of sampling informative pairs in a batch. ProxyNCA [26] proposes an alternative way of modeling the class distribution, where a proxy represents a class, and images in a batch are pushed closer to their corresponding class proxies. Teh et al. [35] built several improvements into the original ProxyNCA method such as fast-moving proxies. In this paper, we further improve on ProxyNCA++ by better capturing the boundaries of the class distribution. Our loss function, ProxyNCAHN++, explicitly samples a set of hard negative images per class in each batch so as to better separate each class in feature space.

3. System Design

Our overall system follows the general paradigm in open-set logo recognition. First, a general logo-region de-

tector such as [3, 28] identifies axis-aligned bounding boxes likely to contain a logo region. Then, a feature embedder module extracts a representation of each bounding box and matches it against a stored image representation of possible logo classes. The two core contributions of this work are in pre-training the feature embedder module to imbue text sensitivity and in using a better loss function for training the feature embedder model. Figure 1 shows the high-level system design. Next, we describe the two unique components of our design.

3.1. OCR Sensitive Embedding

We build on a simple yet effective design from CLIP [27] to perform image-text pre-training. A batch of N image-text pairs are fed to separate vision and language processing modules. The two modalities are trained by computing an $N \times N$ matrix of dot products between all possible image-text pairs. Finally, supervision is applied by encouraging the matrix to be diagonally dominant, i.e., ensuring that embeddings of an input pair are closely aligned compared to all other pairings. The sum of cross-entropy losses for each image-text pair generates the training signal for both vision and language backbones. The resulting model, especially the vision backbone, is found to possess strong OCR discovery capabilities. In particular, Appendix E.2 of [27] documents 80.5% accuracy of the visual encoder on the rendered SST-2 NLP dataset. OCR discovery comes from having rich descriptions that reference scene text contained in an image, e.g. a caption with a brand name corresponding to an image containing that brand's logo. Another advantage of this paradigm is the clear separation between the two modalities, which enables the evaluation of each modality independently.

The WIT dataset collected and used to train the CLIP model is not available publicly, but we can still use the model weights for fine-tuning. We also use our own set of product image-title pairs scraped from an e-commerce website as a comparable image-text pre-training set. Our experiments prove that even with a much smaller dataset than WIT used for pre-training, i.e., approximately 20 million product image-title pairs, we obtain an embedder that significantly outperforms one pre-trained with ImageNet data.

3.2. Loss Function

This section describes the primary loss function used and how we can incorporate complex negative samples explicitly.

3.2.1 ProxyNCA++ Loss

One of the recent standard approaches to training deep metric learning models is the Proxy NCA++ loss [35], which improves on ProxyNCA [26]. The similar normalized soft-

max loss proposed in [41] also utilizes proxies. The main idea is to use trainable proxy vectors to represent the ideal embedding of a given class (generalizable to multiple proxies per class in the dynamic assignment case). As such, we no longer need to sample pairs or triplets of data points within a batch as is the case of a standard contrastive or triplet loss. We can train the model as a classification problem by looking at the distance from each image embedding to its corresponding proxy vector.

Let $\mathcal{P}$ be the set of all learned proxy vectors in $\mathbb{R}^n$. For a given batch of images consider a positive class input image $x_i \in \mathbb{R}^{3 \times p_y \times p_y}$ with logo class $y_i \geq 0$, where $p_x, p_y \in \mathbb{N}$ are the input image width and height, respectively. Further let $z_i \in \mathcal{P}$ denote the proxy vector for x_i, i.e. z_i is closest to $f_\theta(x_i)$ among all proxies with class y_i, where $f_\theta : \mathbf{R}^{3 \times p_y \times p_x} \to \mathbf{R}^n$ is the embedding function with learned parameters θ. For $z_1, z_2 \in \mathbb{R}^n$ define

$$d(z_1, z_2) = \left\| \frac{z_1}{\|z_1\|_2} - \frac{z_2}{\|z_2\|_2} \right\|_2^2, \qquad (1)$$

and let $g(z_1, z_2) = \exp\left(-\frac{d(z_1, z_2)}{\sigma}\right)$, where $\sigma > 0$ is a temperature-scaling parameter. The standard ProxyNCA++ loss for input x_i is then

$$L_{\text{ProxyNCA++}} = -\log(P_i) \qquad (2)$$

$$P_i = \frac{g(f_\theta(x_i), z_i)}{\sum_{z \in \mathcal{P}} g(f_\theta(x_i), z)}. \qquad (3)$$

Note the squared exponent in (1) just as in [35, 26], which we also observed performs better than the standard L^2 distance.

3.2.2 Handling Hard Negatives

In attempt to improve performance further, we consider emphasizing difficult negatives during training that belong to another positive class or no class at all. We employ a batch sampling strategy where for each class y_i, a randomly selected difficult negative class y_j is also included. Then in the denominator of P_i from (3), we can add the exponential distances from the embedding $f_\theta(x_i)$ to all embeddings for class y_j in the batch. To make this more precise, we focus on the closed-set regime where test classes are the same as those in the training set. Let $\mathcal{L} = \{y_1, \ldots, y_k\}$ be the set of all distinct classes, and let $\widetilde{\mathcal{L}} = \{\widetilde{y}_1, \ldots, \widetilde{y}_k\}$ be the set of all (background) negative class labels. Assume we have a mapping $h : \mathcal{L} \ni y_i \mapsto \{h(y_i)_1, \ldots, h(y_i)_{s_i}\} \subset \mathcal{L} \cup \widetilde{\mathcal{L}}$ that assigns each label y_i a set of difficult hard negative classes (possibly empty). Now define for each input image x_i with positive class $y_i \in \mathcal{L}$ in a batch $\mathcal{B}$ the ProxyNCAHN++ loss:

$$L_{\text{ProxyNCAHN++}} = -\log(P_i^{HN}) \qquad (4)$$

where

$$P_i^{HN} \propto \frac{g(f_\theta(x_i), z_i)}{\sum_{z \in \mathcal{P}} g(f_\theta(x_i), z)} \\ + \sum_{x_j \in \mathcal{B} \,|\, y_j \in h(y_i) \cap \mathcal{L}} g(f_\theta(x_i), f_\theta(x_j)) \\ + \sum_{x_j \in \mathcal{B} \,|\, y_j \in \widetilde{\mathcal{L}}} g(f_\theta(x_i), f_\theta(x_j))}. \quad (5)$$

Figure 1(b) shows a conceptualization of the proposed loss function.

3.2.3 Hard Negative Sampling

We utilize the modified loss function (5) by first training a base embedding model using (3) on OpenLogoDet3K47 in the closed-set regime and then computing its confusion matrix C on the validation data, normalized along each row. Then we construct the mapping $h : \mathcal{L} \to \mathbb{P}(\mathcal{L})$ by iterating over the rows i of C, and setting $h(y_i)$ to be the set of all classes y_j such that $\alpha_1 \leq C_{i,j} \leq \alpha_2$. Through validation we determined $\alpha_1 = 0.05$ and $\alpha_2 = 0.35$ to avoid both choosing too-easy negatives and choosing classes y_j that may be visually identical due to label noise. As a secondary filter to avoid nearly identical classes, e.g. atari_1 versus atari_2, we only kept y_j if the levenshtein distance $l(y_i, y_j) > 2$. Once we have constructed such a mapping h, we use a batch sampling strategy as follows. For each batch we sample k positive classes and m samples per class. Then for each class l present, we choose the first random class $\widetilde{l} \in h(l)$ that is not already in the batch, if available. We then sample an additional m images with class $\widetilde{l}$, yielding a total batch size between km and $2km$. We investigate how this approach performs in Section 4.4.

Note that in the above scenario all hard negatives are themselves positives for some other class. We could have also fed a large set of images through a logo-agnostic detector, generated embeddings for each detected region, and then mined the resulting regions sufficiently close to known positives for challenging negative examples that may not belong to any positive class, i.e. that belong to $\widetilde{\mathcal{L}}$. However, this requires more annotation and is difficult to scale unless restricted to find hard negatives for a (small) subset of (challenging) classes.

4. Experiments

4.1. Datasets

There are numerous public datasets pertaining to logo recognition, and we mention those relevant to this work. We also describe the construction of a combined dataset called OpenLogoDet3K47 that we used primarily for ablation studies (see Section 4.4). In general, we remove any images from these datasets with a minimum side length of less than 10 pixels. See Table 1 for information on the main datasets considered. When computing performance, we primarily use the recall@1 metric which is defined as the number of images in a query set identified correctly when retrieving their closest neighbors in a reference set. We omit computing recall@k with $k > 1$ as in [21] since visually similar logos often belong to very different types of brands due to trademark legislation, and thus practically, not retrieving the correct logo class via the closest neighbor would be considered a failure. If a query image also exists in the reference set, then we take the second closest neighbor.

4.1.1 QMUL-OpenLogo

The QMUL-OpenLogo dataset [34] is itself constructed from 7 existing public datasets: FlickrLogos-27 [16], FlickrLogos-32 [30], Logos-32plus [2], BelgaLogos [15], WebLogo-2M [33], Logos In The Wild [38], and SportsLogo [23]. The authors combined these datasets by combining some same-brand classes that exist in multiple sources, removing many erroneous annotations, and removing classes with fewer than 10 images. The result is a relatively high quality dataset containing 27189 images across 309 classes with object-level annotations.

When evaluating on this dataset with a model trained only on LogoDet3K, we closely follow [21] and partition it into a query set, consisting of 10 random images per class, with the remaining images placed in a gallery set. We also partition with respect to whether a class name contains "-text" to provide an insight into performance on text-dominant logo classes. Finally, we compute recall@1 over the entire dataset versus itself for comparison.

4.1.2 LogoDet3K

The LogoDet3K dataset [39] consists of 158652 images with 194261 bounding boxes across 3000 logo classes. When training on this dataset, we use an open-set approach so that test classes are unseen. Furthermore, we place any class with less than 20 samples entirely in the training split. When evaluating on this dataset, just as with OpenLogo we construct a query and gallery set. However, this dataset does not specify text-dominant classes, so we cannot partition further.

4.1.3 FlickrLogos-47

FlickrLogos-47 [30] is an expanded version of FlickrLogos-32, where the same images are used but with more complete annotations that cover additional classes.

esso_text   fosters_text   home_depot_text   mercedesbenz_text

pepsi_text1   santander_text   singha_text   wellsfargo_text

Figure 2. Explicitly marked text-dominant classes from the test split of OpenLogoDet3K47

4.1.4 OpenLogoDet3K47

In order to improve the diversity of classes and volume of training data used in our experiments, we combine the LogoDet3K dataset with two other public datasets that have a relatively small overlap with it: QMUL-OpenLogo and FlickrLogos-47. We cleaned the resulting union of datasets by merging or removing some redundant classes, and details can be found in the supplementary material. After removing image regions with a minimum side length less than 10 pixels and any classes with fewer than 20 instances, we have a total of 2714 classes with 181552 images and 227176 objects.

Similar to OpenLogo and LogoDet3K, we evaluate on the test split by separating it into query and gallery subsets. In particular, we first split the dataset into a global query and gallery set. We further split the query set into large images (greater than 70 pixels on minimum side) and small images. The value 70 was chosen because it is near the 40th percentile of minimum side length values across the entire dataset. Finally, we separate small/large query subsets into those classes with "_text" in the name and those without. Figure 2 shows examples of such text-dominant classes from the test set. This separation facilitates evaluation of recall@1 performance across specific parts of the test dataset, which in our evaluations include *query vs gallery*, *all vs all* (query plus gallery versus itself), *small/large query vs gallery*, and *text query vs gallery*. Note that many classes without "_text" in the name are still text-heavy, so results must be interpreted carefully.

4.1.5 E-commerce Image-Text Pair Dataset

We curated approximately 20 million image-text pairs from an e-commerce website, each containing a main product image and its corresponding product title. This yields a similar dataset to WIT or Laion400m [32] but in a different image domain. We then trained a CLIP model on this dataset to obtain an alternative image-text pre-trained ViT base model to fine-tune for logo recognition.

4.2. Implementation

All experiments were conducted on machines with 4 Tesla V100 GPUs, using PyTorch version 1.10. The training code was built on the Pytorch Metric Learning Library, version 0.9.99. Three different architectures were used: a ViT with patch size 32 (ViT/B-32 [27]); a CLIP-modified ResNet50 architecture (CLIP-ResNet50) whose final pooling is replaced with multi-head attention; a standard ResNet50 architecture from torchvision with a modified final pooling and layer norm before output [25, 35].

The output embedding has 128 dimensions for all models, which we obtain by attaching a fully connected (FC) layer of shape $128 \times d_0$, where d_0 is the output dimension of the base (trunk) architecture. We initialize the proxy vectors by evaluating the pre-trained model (with added fully connected layer) on the training set and computing per-class mean embeddings. All networks were adapted to process input images with 160×160 pixels, which is comparable with input image sizes used in [21, 9]. This is accomplished for ResNet50 by replacing the final average pool layer with an adaptive max pooling layer. For the ViT and modified ResNet50 models used by CLIP, we apply bicubic interpolation to the positional embedding tensors.

Common hyperparameters used are influenced by referencing those cited in [27, 35]. Hyperparameter tuning was conducted on a validation set with approximately 16% of logo classes. The training set comprises approximately 64% of classes, with the remaining 20% of classes assigned to the test set. Any collected negative samples with class in $h(y) \cap \mathcal{L}$ were included in the split containing all positives for class y. For each model architecture and pre-trained weights considered, we conducted 20 trials using the Ax library to select the trunk base learning rate $\gamma_{\text{trunk}} \in [10^{-6}, 10^{-4}]$, final fully connected layer learning rate $\gamma_{\text{fc}} \in [0.0001, 0.01]$, proxy learning rate $\gamma_p \in \{1, 2, \ldots, 100\}$, and batch size $b \in \{128, 192\}$. Each trial was trained for 25 epochs using the AdamW optimizer with learning rates reduced on plateau (patience of 4 and factor of 0.25). The temperature scaling parameter σ in (3) and (5) was fixed at 0.06. Image augmentation used consisted of color jitter and perspective transformations. Other fixed hyperparameters are listed in the supplementary material. The best performing hyperparameter set for each model configuration was used to train a final model on the training and validation sets combined, with evaluation metrics then computed on the test set. In order to build a ViT pre-trained on E-commerce image-text data, we used the same hyperparameters as specified in [27]. We randomly initialize the vision-text backbone of the CLIP model based on ViT and train with a contrastive loss.

Table 1. Public Logo Dataset Comparison

Dataset	Logos	Images	Objects	Annotation	Text/Sym. Sep.
BelgaLogos [15]	37	1321	2697	Object-Level	✓
LogosInTheWild [38]	871 (786)	11054	32850 (25311)	Mixed	✓
FlickrLogo-32 [30]	32	2240	3404	Object-Level	✓
FlickrLogo-47 [30]	47	2235	5968	Object-Level	✓
QMUL-OpenLogo [34]	352	27083	51207	Object-Level	✓
LogoDet3K [39]	3000	158652	194261	Object-Level	
OpenLogoDet3K47	3276 (2714)	188244 (181552)	235738 (227176)	Object-Level	Partial

4.3. Comparison With State-Of-Art On Public Datasets

In Table 2, we see the performance of our best image-text pre-trained model on several unseen public datasets, compared to the previous state-of-art [21] that was also trained on the LogoDet3K dataset. Our model uses 160 pixel input size compared to 150 for the comparison model, and we use 128-dimensional embeddings instead of 512 (1024 with combined text recognition and image features). We trained on the same percentage of logo classes from LogoDet3K as in [21] (80%), but there are no standard splits for this dataset, so we could only ensure that we sampled and prepared the data as closely as possible. In the case of OpenLogo and the LogoDet3K test set, we again mimic [21] and split each into query and gallery sets to give more realistic estimates of zero-shot performance. We show text-specific recall@1 where possible. Ultimately, we see a roughly 3-6% improvement in recall@1 across all subsets where exact comparison data is available, except for BelgaLogo, which can likely be attributed to its small size.

4.4. Ablation Study

We now consider how architecture and pre-training method affect overall performance. We observe in Table 3 that both image-text pre-trained models (either from OpenAI or trained on E-commerce image-title pairs) yield nearly 1% better recall@1 performance (test set queried against itself) than either ImageNet1K-pre-trained model. Furthermore, the recall@1 performance with queries restricted to text-dominant classes has a 2-3% improvement. We show in Figure 3 several failure cases of the best ImageNet1K-pre-trained ViT model relative to the best OpenAI image-text pre-trained ViT model. In particular, we see that the image-text pre-trained model retrieves the correct logo for text-heavy classes like 'Fosters', 'Konka', or 'Ultra Brite', where the font style/color/background differs, while the ImageNet pre-trained retrieves an incorrect match. We also see evidence of OCR sensitivity to non-English (Chinese) characters as evidenced by correct matches in rows 6-8 of Figure 3.

Regarding the effect of loss function (5), we observe in Table 4 that overall recall@1 improves slightly by 0.1%. To see this we trained two models in the closed-set setting

Figure 3. Sample images from the OpenLogoDet3K47 dataset test split showing that our image-text pre-trained ViT logo embedder is better at representing text-dominant logo classes when compared to the same architecture pre-trained on ImageNet. Columns 1+4: query image, columns 2+5: ImageNet-pre-trained ViT model's incorrect retrieval, columns 3+6: Image-text pre-trained ViT model's correct logo retrieval

(i.e. all classes seen during training) using (3) or (5), while holding all other hyperparameters the same.
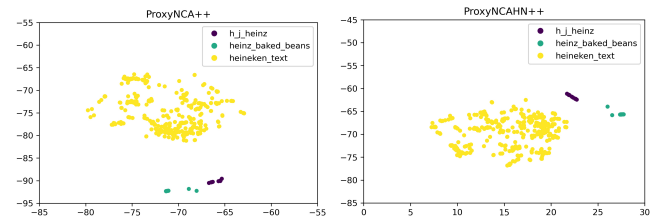

Figure 4. TSNE comparison for test images from hard negative cohort {h_j_heinz, heinz_baked_beans, heineken_text} using models trained with ProxyNCA++ versus ProxyNCAHN++ loss functions.

Table 2. Recall@1 performance on various public logo datasets for the best ViT base model trained on LogoDet3K vs previous state-of-art taken from Table 4 of [21].

Model	LogoDet3K Test		OpenLogo			BelgaLogo		FlickrLogos-47		LiTW	
	Q vs G	All	Q vs G	All	Text	All	Text	All	Text	All	Text
ViT IT Pre-trained	**0.9836**	0.9886	**0.9371**	0.9629	0.9568	**0.9797**	0.9784	**0.9834**	**0.9778**	**0.9391**	0.9456
SeeTek [21]	0.9490	-	0.8964	-	-	0.9739	**1.0**	0.9188	0.9400	0.8772	-

Table 3. Recall@1 performance on the OpenLogoDet3K47 test split among several model initializations. *Pre-training* denotes which data the base model was pre-trained on. For example, *OpenAI IT* refers to OpenAI's image-text pre-trained ViT-B/32 model. *BS* is batch size; *LR (T, E, P)* denotes the trunk, embedder, and proxy learning rates, respectively; *All vs All* denotes recall@1 when querying the entire test dataset against itself and taking the 2nd closest neighbor; *Q vs G* refers to recall@1 for disjoint query and gallery subsets of the test set.

Model	Pre-training	BS	LR (T,E,P)	NMI	All vs All	Q vs G	Text	Small (QvG)	Large (QvG)
ViT-B/32	OpenAI IT	192	(2.3E-06, 0.0015, 71)	**0.9095**	**0.9867**	**0.9788**	**0.9925**	**0.9702**	**0.9843**
ViT-B/32	E-comm. IT	192	(5.0E-06, 0.0015, 70)	0.9040	0.9849	0.9759	0.9888	0.9664	0.9819
CLIP-ResNet50	OpenAI IT	128	(5.3E-06, 0.0005, 67)	0.8854	0.9783	0.9641	0.9716	0.9518	0.9720
ResNet50	ImageNet1K	128	(0.00011, 0.00045, 54)	0.8742	0.9756	0.9551	0.9765	0.9518	0.9573
ViT-B/32	ImageNet1K	128	(3.3E-05, 0.0023, 100)	0.8707	0.9755	0.9523	0.9790	0.9504	0.9536

Table 4. Recall@1 performance on the test split of OpenLogoDet3K47 dataset using loss function (3) vs (5). The model was generated by fine-tuning the OpenAI image-text pre-trained ViT model on the (closed-set) training and validation splits of OpenLogoDet3K47 (i.e. all classes are seen during training)

Model	Pretraining	BS	LR (T,E,P)	Loss	NMI	All vs All	Q vs G	Text	Small (QvG)	Large (QvG)
ViT-B/32	OpenAI IT	192	(3E-06, 0.0003, 63)	(3)	0.9325	0.9588	**0.9480**	0.9341	**0.9340**	0.9583
ViT-B/32	OpenAI IT	192	(3E-06, 0.0003, 63)	(5)	**0.9339**	**0.9600**	0.9475	**0.9366**	0.9312	**0.9594**

Furthermore, looking at the hard negative set mapping h_j_heinz $\rightarrow$ $h($h_j_heinz$)$ = {heinz_baked_beans, heineken_text}, we observe that the corresponding normalized confusion submatrix slightly improves as shown in (6). Also, in Figure 4 the model trained with (5) has qualitatively better separation of h_j_heinz from heinz_baked_beans.

$$\begin{matrix} \text{h\_j\_heinz} \\ \text{heinz\_baked\_beans} \\ \text{heineken\_text} \end{matrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.909 & 0.045 \\ 0 & 0 & 0.979 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.954 & 0 \\ 0 & 0 & 0.966 \end{pmatrix}. \quad (6)$$

Figure 5. Logo retrieval errors showing different challenges for future work

5. Conclusion

This work demonstrated a simple yet effective solution to embue text sensitivity in a logo matching model, which sets new state-of-the-art on all public logo datasets. Our solution leverages image-text multimodal pre-training to discover OCR capabilities which proves vital on text-heavy logo images. Notably, the same model continues to outperform the existing state-of-the-art matching model even on non-text logos. The new proposed ProxyNCAHN++ loss function improves separability on top of a strong baseline. In the future, we will study a better integration of the logo-agnostic detector to understand how upstream detection accuracy influences downstream precision. Further challenges also remain regarding poorly aligned logo bounding boxes in the context of a logo detector/embedder pipeline, blurry/tiny logo regions, poor lighting conditions, partial text matches, ambiguous or indistinct logos, and stylized or non-English text (see Figure 5).

References

[1] Muhammet Bastan, Hao-Yu Wu, Tian Cao, Bhargava Kota, and Mehmet Tek. Large scale open-set deep logo detection, 2022.

[2] Simone Bianco, Marco Buzzelli, Davide Mazzini, and Raimondo Schettini. Deep learning for logo recognition. *Neurocomput.*, 245(C):23–30, jul 2017.

[3] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[4] F. Cesarini, E. Francesconi, M. Gori, S. Marinai, J.Q. Sheng, and G. Soda. A neural-based architecture for spot-noisy logo recognition. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, volume 1, pages 175–179 vol.1, 1997.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] D.S. Doermann, E. Rivlin, and I. Weiss. Logo recognition using geometric invariants. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, pages 894–897, 1993.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[9] István Fehérvári and Srikar Appalaraju. Scalable logo recognition using proxies. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 715–725, 2019.

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[11] Raia Hadsell, Sumit Chopra, and Yann 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. IEEE, 2006.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[13] Changbo Hu, Qun Li, Zhen Zhang, Keng-hao Chang, and Ruofei Zhang. A multimodal fusion framework for brand recognition from product image and context. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–4. IEEE, 2020.

[14] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[15] Alexis Joly and Olivier Buisson. Logo retrieval with a contrario visual query expansion. In *MM '09: Proceedings of the seventeen ACM international conference on Multimedia*, pages 581–584, 2009.

[16] Y. Kalantidis, LG. Pueyo, M. Trevisiol, R. van Zwol, and Y. Avrithis. Scalable triangulation-based logo recognition. In *in Proceedings of ACM International Conference on Multimedia Retrieval (ICMR 2011)*, Trento, Italy, April 2011.

[17] Yash Kant, Dhruv Batra, Peter Anderson, Alexander Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multimodal transformers for textvqa. In *European Conference on Computer Vision*, pages 715–732. Springer, 2020.

[18] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. *Advances in Neural Information Processing Systems*, 33:2611–2624, 2020.

[19] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] Chenge Li, István Fehérvári, Xiaonan Zhao, Ives Macedo, and Srikar Appalaraju. Seetek: Very large-scale open-set logo recognition with text-aware metric learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2544–2553, January 2022.

[22] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34, 2021.

[23] Yuan Liao, Xiaoqing Lu, Chengcui Zhang, Yongtao Wang, and Zhi Tang. Mutual enhancement for detection of multiple logos in sports videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4856–4865, 2017.

[24] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[25] Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. *Proceedings of the 18th ACM international conference on Multimedia*, 2010.

[26] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 360–368, Los Alamitos, CA, USA, oct 2017. IEEE Computer Society.

[27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.

[28] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[30] Stefan Romberg, Lluis Garcia Pueyo, Rainer Lienhart, and Roelof van Zwol. Scalable logo recognition in real-world images. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, ICMR '11, New York, NY, USA, 2011. Association for Computing Machinery.

[31] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[32] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs, 2021.

[33] Hang Su, Shaogang Gong, and Xiatian Zhu. Weblogo-2m: Scalable logo detection by deep learning from the web. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 270–279, 2017.

[34] Hang Su, Xiatian Zhu, and Shaogang Gong. Open logo detection challenge. In *British Machine Vision Conference*, 2018.

[35] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. *arXiv preprint arXiv:2004.01113*, 2020.

[36] Osman Tursun and Sinan Kalkan. Metu dataset: A big dataset for benchmarking trademark retrieval. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, pages 514–517. IEEE, 2015.

[37] Andras Tüzkö, Christian Herrmann, Daniel Manger, and Jürgen Beyerer. Open set logo detection and retrieval. *arXiv preprint arXiv:1710.10891*, 2017.

[38] Andras Tüzkö., Christian Herrmann., Daniel Manger., and Jürgen 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 - Volume 5: VISAPP,*, pages 284–292. INSTICC, SciTePress, 2018.

[39] Jing Wang, Weiqing Min, Sujuan Hou, Shengnan Ma, Yuanjie Zheng, and Shuqiang Jiang. Logodet-3k: A large-scale image dataset for logo detection. *ACM Trans. Multimedia Comput. Commun. Appl.*, 18(1), jan 2022.

[40] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. Tap: Text-aware pre-training for text-vqa and text-caption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8751–8761, 2021.

[41] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning, 2019.

[42] Qi Zhu, Chenyu Gao, Peng Wang, and Qi Wu. Simple is not easy: A simple strong baseline for textvqa and textcaps. *arXiv preprint arXiv:2012.05153*, 2, 2020.