

# DEEP LEARNING FOR LOGO MINING

*Saurabh Kumar, Huy Nguyen*

Georgia Tech, Yahoo

## ABSTRACT

Logo recognition and localization in images has a multitude of business applications, particularly for advertising or trademark protection purposes. Existing methods to create datasets that are used for effective logo detection require a lot of human input to identify images which contain logos and specify where the logos are located in the selected images. In this paper, we propose a novel method to mine logos in images in a more autonomous way. We train a convolutional neural network to determine whether or not any logo is present in a given image. This network is used to filter a large set of candidate images which may or may not contain logos. We then employ Exemplar SVMs with convolutional neural network features to produce bounding box annotations for images that contain logos.

**Index Terms**— logo detection, logo mining, convolutional neural networks, Exemplar SVMs

## 1. INTRODUCTION

Logo recognition has a variety of industrial and business applications. Accurate recognition of logos, for example, can improve contextual advertising. Many advertising platforms rely purely on the use of natural language processing and perhaps some basic information, while brands within images may be completely ignored [1]. One bottleneck of robust computer vision based logo detection is the lack of many large-scale logo datasets that can be used as training data for logo recognition and localization models. Perhaps the most well-known logo dataset is "FlickrLogos-32," which contains 32 logo classes and 8240 logo images [2]. Recently, a much larger scale database for logo detection, called "Logo-Net," was constructed for such applications [3].

Industrial logo detection is often concerned with the detection of logos "in the wild." Logos in the wild are logos which appear in real-world images; these logos are more difficult to classify and localize than canonical logo images due to transformations such as rotation and translation, as well as the effects of illumination and occlusion. These transformations pose challenges, not only in logo recognition and localization, but also in the creation of logo datasets that contain such images. For example, in the construction of Logo-Net, human annotators were required for classifying logos in

images and producing bounding box annotations [3]. When creating datasets which contain hundreds of thousands of images across many different logo classes, the cost of acquiring human annotations through avenues such as crowdsourcing can be expensive. In this paper, we present a method through which we use deep learning techniques to mine logos nearly autonomously. Therefore, our approach allows for logo dataset construction in a more cost efficient manner.

## 2. FLICKRLOGOS-32 DATASET

All deep neural network architectures in our logo mining approach are trained on the FlickrLogos-32 dataset. This dataset contains images with logos that have bounding box annotations to identify where in the image the logo is located, as well as images which do not contain any logos. We use this dataset to train a logo-nologo classifier and our Exemplar-SVM method, which are described below.

## 3. LOGO MINING APPROACH

In this section, we discuss the procedure we develop to facilitate the construction of a large logo dataset with little human input required for logo class identification and bounding box annotation. This approach can be used to create a large logo dataset containing images of logos in the wild so that industrial logo detection is possible. We develop a pipeline to search for images with desired logos, filter out images not containing logos from the returned results, and produce bounding box annotations for the filtered images. The goal is to design this pipeline so that an object detection framework, such as Faster R-CNN [4] or the Single Shot Multibox Detector [5], can be trained on a dataset constructed using our approach to perform real-time logo detection. We perform our experiments using the Caffe deep learning framework [6].

### 3.1. Data Collection and Filtering

To determine which brands are important for industrial logo detection, we acquire a list of 500 brands [7]. We use the brand names in the list as queries to the Flickr API to obtain URLs for images that contain the logo(s) associated with the brand. Because the search items used are generic, a significant portion of the results are images which do not contain

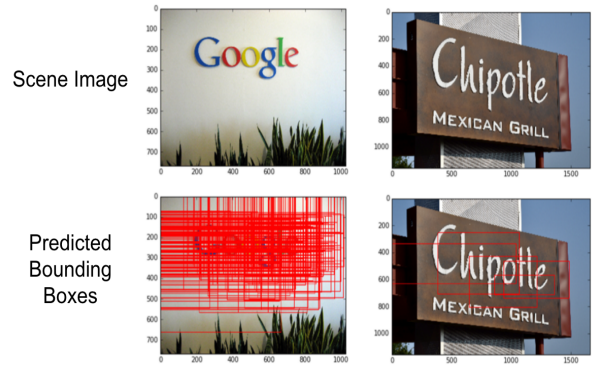
logos in them. For example, searching "Apple" returns images with apples in addition to those containing Apple logos. The purpose of using generic search terms rather than more specific queries, such as "Apple Logo," is to obtain as many images of logos in the wild as possible rather than canonical logo images. We obtain up to 100 images per month from January 2005 to May 2016 via the Flickr API for each brand query. Performing this search yields 42 million image URLs.

The majority of the 42 million images do not contain logos due to the reasons described above. We develop an approach to filter the 42 million images that are obtained through performing generic searches. We train a GoogLeNet network on the FlickrLogos-32 dataset to output whether or not a given image contains a logo. The FlickrLogos-32 dataset has 33 classes: 32 logo classes and one no-logo class. We condense the 32 logo classes into one label that indicates that the image contains a logo. We then train a GoogLeNet network pretrained on ImageNet on the combined *trainval* and *test* sets using the modified labelling for 100k iterations with batch size 32, learning rate of 0.001, and momentum of 0.9. The softmax loss layers of the network are set to have two outputs for the probabilities that the image does or does not contain a logo. We refer to this trained GoogLeNet network as the logo-nologo classifier.

We validate the logo-nologo classifier on a small dataset that we create by randomly sampling 100 images from the 42 million generic search results and annotating each image by hand as either containing or not containing a logo. The logo-nologo classifier achieves 77.8% accuracy on this dataset, with a false positive rate of 0.1 and a false negative rate of 0.58. We apply the logo-nologo classifier to the total 42 million generic search results and throw out images which the classifier outputs as not containing a logo, resulting in a filtered dataset with 7.5 million images. In the set of filtered images, each image's class label is its associated brand query. We throw out brand classes for which there are fewer than 250 images, resulting in 392 logo classes.

### 3.2. Bounding Box Annotations

We develop methods to further filter the 7 million images while producing good quality bounding boxes around a logo if it exists in an image. Our goal is to make the bounding box annotation process autonomous so that conventional crowd-sourcing based approaches, such as the use of Mechanical Turk, are not required. We experiment with two approaches: similarity-based dense sampling and Exemplar SVM based dense sampling. In both approaches, we find a "template" image of the logo we wish to find in images. This template image is a canonical logo image which contains the logo on a white background, and it is cropped to minimize the number of background pixels in the image. We construct convolutional neural network based features for each template image by inputting the image into a GoogLeNet network [8] pre-



**Fig. 1.** Predicted Bounding Boxes for Google and Chipotle Scene Images. Using a threshold of 0.95, the selected bounding boxes for a Google scene image and a Chipotle scene image are shown. This figure displays the bounding boxes prior to non-maximum suppression.

trained on the ImageNet dataset [9] and extracting the output of the pool5/7x7\_s1 layer. This process produces a  $1 \times 1024$  feature vector for each template image.

### 3.3. Similarity Based Dense Sampling

The similarity-based dense sampling procedure entails searching for regions containing the desired logo in an image that is not our template image, denoted as a "scene" image, and constructing a bounding box based on a non-maximum suppression scheme of these regions. To determine which regions to search, we densely sample the image using a sliding window with aspect ratios 1, 1.5, 2, 2.5, 3, 3.5, 4, and their reciprocals. The various aspect ratios allow the sliding windows to better capture logos of various shapes in the scene images. For example, the aspect ratio of a Google logo is different from that of an Apple logo.

We employ a Gaussian pyramid to search for logos of different scales; for example, a small logo in an image is found by upsampling the image and applying the sliding window densely across the upsampled image. For each sliding window, we compute a  $1 \times 1024$  feature vector using the same approach used for the template image. We threshold the windows based on the L2-distance of their features to the feature vector computed using the template image. We select any windows below our threshold as our predicted bounding box outputs. We then perform non-maximum suppression on the predicted bounding boxes by sorting them by their L2 distances, which we refer to as "scores," and suppressing bounding boxes that overlap a better scoring bounding box by greater than a preset threshold. The jaccard overlap is computed as the intersection of the two bounding boxes divided by their union.

A problem with this approach is that it is difficult to determine apriori which threshold to use, since the optimal thresh-










olds vary between different brands. As shown in Figure 1, using a particular threshold yields different results for an image with a Google logo and an image with a Chipotle logo. A potential solution is to select a fixed number of boxes with the best scores rather than using a threshold, but this results in bad quality bounding boxes selected in some images.

### 3.4. Exemplar SVM with ConvNet Features

We address the problem of finding an optimal threshold with the similarity-based dense sampling method by using the Exemplar SVM approach to object detection developed by Malisiewicz et al [10]. This method entails training a linear SVM classifier with one positive example and many negative windows. A histogram of oriented gradients (HOG) template is used as the features for the positive example. The trained SVM is then used to identify instances that are visually similar to the positive example. However, rather than representing the example with a HOG template, we compute a GoogLeNet feature vector as described above for the L2 similarity method. We employ Exemplar SVM style training using a ConvNet-based feature representation. For each brand, we pick a canonical image of the associated logo as the positive example, and we apply a sliding window through a Gaussian image pyramid for each of 100 randomly selected Flickr images to obtain negative examples. We train a separate linear SVM for each positive example, which uses as training data one positive example and thousands of negative windows. We obtain the negative windows from 100 randomly sampled Flickr images that do not contain any logos; we randomly sample around 1000 windows from each image by applying a sliding window at different scales.

In order to bypass the need to search for an optimal threshold for the SVM output, we calibrate the trained classifier so that we can use a fixed threshold of 0.5 across all brands. For each brand, we sample an image containing the associated logo in the wild and draw a bounding box around the logo in the image. We refer to such an image as the calibration image for the given logo class. Multiple bounding boxes are drawn if the logo appears more than once in the image. We then apply a sliding window across an image pyramid for this image, and compute the GoogLeNet feature vector and SVM score for each window. We label each window which has a jaccard overlap of over 0.5 with a ground truth bounding box as the positive class, +1, and each window which has a jaccard overlap of less than 0.2 with a ground truth bounding box as the negative class, -1.

After constructing the calibration data using the procedure described above, we fit a logistic regression function to the resulting windows where the features are the SVM scores and the label for each window is 1 or -1. For any image in which we wish to draw a bounding box around the brand's logo, we apply the sliding window across a Gaussian image pyramid, compute a GoogLeNet feature vector for each window, out-

| Brand  | Template Image  | Calibration image   | Sample Result   |
|--------|---|---|---|
| Google |  |  |  |
| Apple  |  |  |  |
| Shell  |  |  |  |

**Fig. 2.** Predicted bounding boxes for three different logo classes (sourced from FlickrLogos-32). This displays the acquired bounding boxes using our heuristic after postprocessing.

put an SVM score for each feature vector using the trained SVM classifier, and then input the SVM score into the logistic regression function to output a probability, which we treat as a calibrated score. We threshold this calibrated score at 0.5. A score above 0.5 indicates that the window is a candidate bounding box and score of at most 0.5 means that it is not. We apply non-maximum suppression to the candidate bounding boxes, and then merge any overlapping boxes by computing the smallest bounding box encompassing a set of overlapping bounding boxes.

The Exemplar SVM pipeline we have developed can be employed to acquire the bounding box regions that serve as the ground truth bounding boxes in logo detection datasets. It can also further filter out images that do not contain logos among pre-filtered images, such as the 7 million images that were filtered using the logo-nologo classifier, by outputting zero candidate bounding boxes for those images. See Figure 2 for some sample results for different logo images. The Exemplar SVM pipeline is a data efficient and automated process, as we only require one template image, one calibration image with annotated ground truth bounding boxes, and a set of negative images to obtaining bounding boxes for logos in a wild for a particular brand.

### 3.5. Exemplar SVM Experiments

In order to determine the viability of using Exemplar SVMs for producing bounding box annotations, we validate our approach using the FlickrLogos-32 dataset. We design three experiments in which we test the Exemplar SVM approach against the ground truth bounding boxes in the FlickrLogos-32 *test* set for each of the 32 logo classes. For all experiments, we randomly select an image along with its bounding box annotation from the FlickrLogos-32 *trainval* set for each logo class to use as the calibration image. In Experiment 1, we use one canonical logo image as the positive example for

|           | adidas<br>bmw<br>carls<br>chimay | coke<br>corona<br>dhl<br>esso | fedex<br>ferrari<br>ford<br>fosters | google<br>guinness<br>heineken<br>HP | milka<br>nvidia<br>shell<br>singha | starbucks<br>texaco<br>tsingtao<br>ups |
|-----------|----------------------------------|-------------------------------|-------------------------------------|--------------------------------------|------------------------------------|--|
| Precision | 0.80                             | 0.87                          | 0.79                                | 0.51                                 | 1.0                                | 0.95                                   |
|           | 0.86                             | 0.20                          | 0.87                                | 0.73                                 | 0.26                               | 0.80                                   |
|           | 0.37                             | 0.58                          | 0.82                                | 0.45                                 | 0.16                               | 0.50                                   |
|           | 0.56                             | 0.87                          | 0.74                                | 0.34                                 | 0.25                               | 1.0                                    |
| Recall    | 0.66                             | 0.66                          | 0.72                                | 0.88                                 | 0.05                               | 1.0                                    |
|           | 0.83                             | 0.37                          | 0.94                                | 0.64                                 | 0.19                               | 0.71                                   |
|           | 0.55                             | 0.79                          | 0.86                                | 0.37                                 | 0.32                               | 0.40                                   |
|           | 0.38                             | 0.68                          | 0.76                                | 0.34                                 | 0.06                               | 0.33                                   |

**Table 1.** Experiment 3 precision and recall rates across 24 logo classes in the FlickrLogos-32 test set.

each brand. In Experiment 2, we augment the positive data by overlaying each canonical logo on 100 background images to create a total of 100 positive examples for each brand. In Experiment 3, we do not employ any data augmentation, but instead of using the pool5/7x7\_s1 layer to construct features, we extract the output of the pool4/3x3 layer and apply PCA to reduce the dimensionality to 1x1024 so that it is consistent with the feature dimensionality used in Experiments 1 and 2.

In each experiment, we measure the precision and recall rates for the detection of logos for each brand. We threshold the logistic regression output at 0.5 and do not apply any merging or non-maximum suppression. A predicted bounding box is considered a correct prediction if its jaccard overlap with the ground truth bounding box in the given image is at least 0.5; the prediction is incorrect otherwise. For all experiments, the precision and recall rates were below 0.1 for 8 logo classes, and the results of those logo classes are not included. Upon analyzing sources of error, we believe that the canonical logo images selected for those brands are not representative of most of the logos present in the test images. At the time of writing this paper, we have not yet tried different logo templates for those brands. Another possible source of error could be the calibration images that have been selected for these brands. The average precision rates across the remaining 24 logo classes for Experiments 1, 2, and 3 are 0.55, 0.51, 0.64, and the average recall rates are 0.35, 0.38, and 0.54, respectively. See Table 1 for a summary of the results of Experiment 3.

#### 4. CONCLUSION

In this paper, we describe an approach that can be applied to logo dataset construction. The combined filtering and bounding box annotation pipeline allows for more cost efficient logo mining. The use of Exemplar SVMs with convolutional neural network based features shows promising results for bounding box annotations. We leave further refinement of the Exemplar SVM method to improve performance to future work.

The end goal is to use this approach to build a large logo dataset containing a wide variety of logo classes so that robust logo detectors can be built.

#### 5. REFERENCES

- [1] Forrest N. Iandola, Anting Shen, Peter Gao, and Kurt Keutzer, “Deeplogo: Hitting logo recognition with the deep neural network hammer,” *CoRR*, vol. abs/1510.02131, 2015.
- [2] Stefan Romberg, Lluís 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*, New York, NY, USA, 2011, ICMR ’11, pp. 25:1–25:8, ACM.
- [3] Steven C. H. Hoi, Xiongwei Wu, Hantang Liu, Yue Wu, Huiqiong Wang, Hui Xue, and Qiang Wu, “Logo-net: Large-scale deep logo detection and brand recognition with deep region-based convolutional networks,” *CoRR*, vol. abs/1511.02462, 2015.
- [4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in *Advances in Neural Information Processing Systems (NIPS)*, 2015.
- [5] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, “SSD: Single shot multibox detector,” *arXiv preprint arXiv:1512.02325*, 2015.
- [6] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, “Caffe: Convolutional architecture for fast feature embedding,” 2014.
- [7] “Best global brands — brand profiles and valuations of the world’s top brands,” .
- [8] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich, “Going deeper with convolutions,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
- [10] Tomasz Malisiewicz, Abhinav Gupta, and Alexei Efros, “Ensemble of exemplar-svm for object detection and beyond,” in *Proceedings of the 2011 International Conference on Computer Vision*, Washington, DC, USA, 2011, ICCV ’11, pp. 89–96, IEEE Computer Society.