# Logo Detection Using Deep Learning with Pretrained CNN Models

Salma Sahel
College of Computer and Information Systems
Umm Al-Qura University
Makkah, Saudi Arabia
salma.a.sahel@gmail.com

Mashael Alsahafi
College of Computer and Information Systems
Umm Al-Qura University
Makkah, Saudi Arabia
m.e.alsahafi@hotmail.com

Manal Alghamdi
College of Computer and Information Systems
Umm Al-Qura University
Makkah, Saudi Arabia
maalghamdi@uqu.edu.sa

Tahani Alsubait
College of Computer and Information Systems
Umm Al-Qura University
Makkah, Saudi Arabia
tmsubait@uqu.edu.sa

**Abstract—Logo detection in images and videos is considered a key task for various applications, such as vehicle logo detection for traffic-monitoring systems, copyright infringement detection, and contextual content placement. The main contribution of this work is the application of emerging deep learning techniques to perform brand and logo recognition tasks through the use of multiple modern convolutional neural network models. In this work, pre-trained object detection models are utilized in order to enhance the performance of logo detection tasks when only a portion of labeled training images taken in truthful context is obtainable, evading wide manual classification costs. Superior logo detection results were obtained. In this study, the FlickrLogos-32 dataset was used, which is a common public dataset for logo detection and brand recognition from real-world product images. For model evaluation, the efficiency of creating the model and of its accuracy was considered.**

***Keywords—logo detection; deep learning; convolutional neural networks; FlickrLogos-32***

## I. INTRODUCTION

Logo Detection (LD) is important for several real-world applications [1] by enhancing the ability of users/systems to recognize the identities of items by their brand logos. As a result, LD is an area that many companies aim to explore to captivate their clients and make informed decisions regarding brand development [2]. LD is a sub-field of object detection which is the task of identifying wherever there is a specific object in an image. We are not only interested in simply identifying the existence of an object, but also to locate it within the image. There are many classical approaches to object detection such as key point-based detectors and local feature-based recognition. More recently, there has been a growing interest in using Convolutional Neural Networks (CNNs) to perform object detection tasks. This approach usually starts by capturing images by camera devices, probably experimenting with different resolutions, and then processing the images in order to be able to classify the images of the objects contained within [3]. More specifically, three steps can be used in traditional object detection and classification approaches, and these are: informative region selection, feature extraction, and classification [4]. In region selection, it is probable to scan the whole image consuming a multi-scale sliding window, as numerous substances (objects) might appear in different positions with several sizes and feature ratios.

There are differences between LD and Logo Recognition (LR). LR is used to detect a logo's identity in an image, whereas LD is considered to detect a logo's position [5]. Both LD and LR are based on classification methods. For example, [6] proposed the use of CNNs for a recognition pipeline comprising a recall-oriented logo region. A CNN, in general, is considered as a widely used method in both LD and LR. Research in LD has been addressed in several directions where general object detection with deep learning methods has always been an excessive realization. Construction of a deep object detection model classically needs many categorized training data collected using wide manual classification. In this paper, RCNN, FRCNN and RetinaNet were used for logo detection. This paper presents the current LD methods and sheds light on common datasets used in previous research.

## II. LITERATURE REVIEW

One of the aims of this research is to investigate the area of LD and examine the state-of-the-art approaches. It should be noted that due to the desire to explore new trends in LD, which is a very fast-growing area, we focused on papers published during the last five years or papers that have shown a considerable impact in terms of number of citations. Table I shows the summary of the studied and referenced articles.

Corresponding author: Tahani Alsubait

TABLE I. SUMMARY OF THE REVIEWED ARTICLES

| Reference | Year | Approach | Dataset | Performance |
|---|---|---|---|---|
| [7] | 2019 | YOLOv3 | VLD-3 | 76.6-98.8 |
| [8] | 2017 | Faster R-CNN | FlickerLogo | 0.52-0.67 |
| [1] | 2019 | OSLD - SDML | BLAC and Flickr-32 | 84.5-90.9 |
| [9] | 2018 | RetinaNet | INbreast - GURO | 86-1.00 |
| [10] | 2017 | Faster R-CNN | FlickrLogos | mAP 51-66% |
| [11] | 2020 | Faster R-CNN and YOLOv2 | WebLogo-2M | mAP 36.8-46.9 |
| [12] | 2017 | DenseNet16 -, ResNet101 -, VGG16 | FlickrLogos-32 SportsLogos | 0.37-0.46 |
| [13] | 2017 | Scalable Logo Self Training (SLST) | WebLogo-2M | mAP 34.37% |
| [14] | 2007 | ANN and Support Vector Machine (SVM), Fisher classifier | Tobacco-800 | 39.3-84.2% |
| [5] | 2015 | RCNN – FRCN – SPPnet | Logos-18 test set Logos-160 test set | 81.3-95.2% |
| [15] | 2019 | Faster R-CNN, SSD, YOLOv3 | FlickrLogos-32, PL2K | 0.565 mAP |
| [6] | 2017 | CNN | FlickrLogos-32 Logos-32plus | 0.1-95.8% |
| [2] | 2016 | Fast Region-based Convolutional Networks (FRCN) | ILSVRC, FlickrLogos-32 | mAP 54.5-73.74% |
| [16] | 2018 | Retina U-Net, Mask R-CNN, Faster R-CNN +, U-Faster R-CNN, + DetU-Net | LIDC-IDRI | mAP 29-50.5% |
| [17] | 2017 | Faster R-CNN | FlickrLogo-32 TopLogo-10 | mAP 20.5-81.1% |
| [18] | 2020 | DenseNet-CNN | FlickrLogo-32 | 92.8% |

### *A. Logo Detection Methods*

Several frameworks have been proposed in the LD area, and experimental evaluations show different results. For example, in [2], CNNs were shown to be a reliable technique for LD, containing several layers that are similar to simple and complex cells in the primary visual cortex. In particular, it adopts a hierarchical structure that enables the recognition of visual patterns arising from image pixels. The convolutional layers alternate with other subsampling layers that have varying sizes. As a result, they help enhancing image clarity and improve LD. Other researchers [11, 13] indicated that the technique of scalable logo self-co-learning can self-discover informative training pictures from the blaring web information. In particular, this can be helpful to enhance model capacity in a cross-model co-learning approach [13]. The scalability of the logo is of concern because it determines the level of categorization possible and the efficiency of the LD approach. The process enables the collection of many unconstrained images for review. The images need not undergo any fine-grained instance level of labeling for the necessary aspects to become clearer to the users. However, sometimes, the use of proxies for the scalable LR enables the classification and identification of the logos [15]. In particular, this is because there is no clear definition of a logo and logos vary like brands. The proxies, however, facilitate the re-training of images to integrate all variations. Although it may seem impractical, the utilization of proxies in scalable LR involves the use of a universal logo detector and a few-shot logo recognizer. As a result, the use of proxies can become a reliable tool because it can enable the users to enhance clarity and assess the image better. Additionally, companies can utilize the proxies in scalable LR to achieve high precision and make informed decisions.

Authors in [16] used Retina U-Net for semantic separation in classifying medical images. Retina U-Net is a useful tool for logo detection and classification, which can guarantee the desired results for image analysis. Retina U-Net can be a reliable LD and image markup technology. Companies can rely on this technology to enhance the clarity of their brands. They can make brands more attractive and can be easily valued by customers. In this regard, Retina U-Net can become a suitable tool through which companies can attract customers. Authors in [14] proposed an LD method which used a boosting approach through multiple image scales particularly for LD and image extraction. It uses a trained fisher classifier, a kind of deep learning approach, to perform initial classification that can identify features from document context and linked components. In particular, each logo area is classified by a cascade of simple classifiers to continuously improve scales. It is considered to be able to detect regions to be refined and ignores false alarms. Early research produces weak performance of Faster R-CNN [10]. Therefore, the researchers in [8, 10] introduced a Fast RCNN approach which is based on deep neural networks where the convolutional layers are used to extract gradually abstract feature representations by using previous learned convolutions, then apply a non-linear activation task to the image. More recently, authors in [18] proposed a transfer-learning-based method aided by the use of Densely Connected Convolutional Networks (DenseNet) for logo recognition. They applied their proposed method to the FlickerLogos-32 dataset.

In Computer Aided Diagnosis (CAD) systems [9], the system utilizes a mass detection model based on RetinaNet. RetinaNet is a kind of deep CNN, where an object detector is mainly expected as a one-stage object detector that is fast and effective while achieving improved performance. However, new research in this area [11] argues that the current LD approaches typically consider a small number of logo sessions, with limited images per session and presume fine-tuning associated to each object annotation. However, this produces the problem of limited ability to be scalable to dynamic applications in the real world. Their proposed approach tries to overcome these challenges by ignoring manual labeling and directly exploring web data learning principles. In particular, it proposes an incremental learning approach, named Scalable Logo Self-Co-Learning (SL 2). This method can automatically self-discover informative training images from noisy web data for increasingly improving model ability in a cross-model co-learning means.

### *B. Datasets Used*

The datasets that have been used in recent studies come from various sources. Authors in [14] utilized the Tobacco-800 dataset which consists of 42 million pages of documents, whereas authors in [8, 10] have used four publicly available datasets. Authors in [11] introduced a very large logo dataset which contains: 2,190,757 images of 194 logo classes, named “WebLogo-2M” by designing an automatic approach for data collection and processing by automatically sampling web logo images from social media sources (Twitter). In other studies, such as [2, 6, 15], the public dataset FlickrLogos-32 was used. This dataset is one of the most commonly used datasets in the field of logos, it contains 8,248 pictures, with 32 images per brand. Authors in [15] introduced a new logo dataset containing 2000 logos and 295K images collected from Amazon, called PL2K. The dataset used in the current paper is FlickrLogos-32 [19]. This dataset has 32 logos each has many examples and it also has well-articulated annotations. This is the most prevalent logo detection dataset containing and it has been used for the purposes of this research for comparisons with existing approaches in the area. Sample data from this dataset can be seen in [6, 19].

### *C. Evaluation Stage*

Evaluation of the considered methods has been conducted in a variety of ways through several steps. The framework presented in [10] has been evaluated on the dataset in order to improve the detection performance on small object instances. The evaluation of the approach presented in [8] was conducted on the dataset improving the RPN performance from 0.52 to 0.71 (mAP) and the detection performance from 0.52 to 0.67 (mAP). A description of the evaluation metrics will be presented in the following sections. Other researchers [2] used metrics of performance for measuring LD performance such as the Average Precision (AP) for every logo class and the mean AP (mAP) for all classes, where the detection is considered true when the Intersection over Union (IoU) overtakes 50%. In addition, two evaluations per each detector type (single and five shot) were used in [15]. The Faster R-CNN model worked best with a mAP of 0.56558, where the mAP was decided by region proposals with a class detection threshold of 0.5. For example, the proposed models were trained on PL2K and evaluated on FlickrLogos32 to achieve new state-of-the-art performance of 56.55% mAP. In the evaluation stage in [15], the researchers used an evaluation metric based on labeled ground truth to measure the quality of LD. In [11], the evaluation has been conducted by using extensive comparative evaluations demonstrating the superiority of SL 2 over the state-of-the-art contemporary web data learning methods and strongly weakly supervised detection models. The experimental evaluation of [9], shows that the considered model extracts inconsistent mass features from the single dataset as well as the combined dataset whose mammograms are collected from different sources, which proposes the ability of the model to be applied to different groups. The evaluation of the proposed model in [9] has been addressed in setups consuming pre-trained weights, which uses weights pre-trained on GURO, training and testing on INbreast. This shows that consuming the pre-trained weights on datasets produces the same performance as directly consuming datasets in the training stage.

## III. Methodology

The studied dataset has been examined with the use of some pre-trained deep learning models for object detection based on CNNs.

### *A. CNN Models*

#### *1) RetinaNet*

RetinaNet [20] consists of a support network, and binary sub-nets that use various features of the maps of the provision network. One establishment subnet categorizes the instance of the image, and one regression subnet registers out the bounding box. The model workflow is:

- Loading and preparing training data.
- Training a deep neural network using RetinaNet.
- Evaluating the model.
- Using the model for inference.

#### *2) Faster R-CNN*

Faster R-CNN [21] assimilates the area suggestion algorithm into the CNN model. Faster R-CNN model is composed of an RPN (Region Proposal Network) and a firm R-CNN with communal convolutional feature layers. The model workflow is:

- Using the image classification pre-trained model.
- Fine-tuning the RPN for the area proposal task, which is prepared using the pre-trained image classification model where optimistic samples have IoU>0.7, while undesirable samples have IoU <0.3.
    - Slips a small n×n three-dimensional gap over the convention of the feature map of the complete image.
    - At the middle of each gliding window, it forecasts numerous areas of numerous balances and ratios concurrently. A presenter is a grouping of gliding the window center, scale, and ratio. For instance, for 3 scales + 3 ratios, then k=9 anchors at each gliding position.
- Trains the Fast R-CNN LD model by the usage of the proposals formed by the current RPN.
- Uses the Fast R-CNN system to modify the RPN training. Through the observance of the public convolutional layers, solitarily fine tunes the RPN-precise layers. At this level, the RPN and the finding network have public convolutional layers.

#### *3) R-CNN*

R-CNN [22] is one of the most common deep learning frameworks used to detect objects on a large scale. It is a combination of CNNs and region suggestion. The model workflow is as follows:

- Train the CNN network for image classification.
- Propose category-independent regions of interest by selective search.
- The areas of the objects are warped to make sure that they are in place for the fixed size needed by the CNN.
- Move on with the fine-tuning of the CNN which is warped over the proposal areas for K + 1 classes. The extra class is associated with the background (does not have many objects of interest).
- Assure that every image in the area, with one or more of the forwarded flows of movement throughout the CNN makes a feature vector.
- Decrease the localization of the errors.

### *B. Evaluation Measures*

#### *1) Localization and Intersection over Union*

As can be seen in [22], the object detection of the dataset can be estimated accurately by the IoU which is known to be used in the evaluation stage [23]. The predication of locating the object is determined on how strict the model is, by the evaluation of the function of object localization of the model. This is usually accomplished by drawing a bounding box around the desired object [22]. The function of localization is evaluated by the IoU:

$$\mathrm{IoU} = \frac{\text{area (groudtruthboundingbox} \cap \text{predictedboundingbox)}}{\text{area (groudtruthboundingbox} \cup \text{predictedboundingbox)}} > 0.5 \quad (1)$$

Each object is associated with one bounding box, but in some cases, the bounding boxes might be more than one. When there are more than one bounding boxes regarded to one object, one box will either be True Positive (TP) or False Positive (FP) and the other box is vice versa. Yet, an object can be recognized as False Negative (FN) when there is no predicted bounding box.

#### *2) Mean Average Precision*

mAP is one of the most popular metrics in measuring the accuracy of object detectors, e.g. R-CNN. It is also called precision-recall for detecting bounding boxes. Precision-recall is considered to measure how the network understands importance and how it removes invalid information. mAP enhances the information produced by precision-recall. The prediction is more accurate when the mAP score is higher.

$$\mathrm{mAP} = \frac{1}{\mathrm{N}} \sum_{\mathrm{N}}^{\mathrm{i=1}} \mathrm{APi} \quad (2)$$

The AP is the average of class predictions (N) measured over various thresholds, i.e., it is the average of precision values for various recall levels [24]. The current study aligns with the approach in [24]. In particular, in the current study, the AP curves for each logo class are based on the pre-trained models (R-CNN, Faster R-CNN, RetinaNet).

## IV. IMPLEMENTATION

Initially, we started by cleaning the dataset and checking if the annotation files were correct. Then the dataset needed to be fed into each of the training models. Since it takes too much time to build the model, the model needs to be stored and then reused. After that, we chose samples of pictures from the dataset to train each model on them, in order to produce weights. The remaining images were used for testing the models in order to find each one's testing accuracy. For implementation, Colab notebook was used to execute code on Google's cloud servers. Hardware specifications were: CPU: AMD ryzen 7, GPU: 2080ti, Ram: 16GB, GPU capacity needed for the build: 2.3GB for R-CNN, 1.5GB for FR-CNN, and 3.1GB for RetinaNet.

## V. RESULTS AND DISCUSSION

It has been found that R-CNN takes more time and GPU space, but obtained higher accuracy. Hence, the increased accuracy comes with a cost. This is because the FR-CNN first applies CNN and then the zones where compared to the R-CNN which makes the regions first and then applies CNN. When comparing the R-CNN and the FR-CNN to the RetinaNet, it was revealed that the RetinaNet takes fewer test time when compared to R-CNN, it demands more training time when compared to the FR-CNN and it takes more space than both R-CNN and FR-CNN because it checks layers more, because the quality of the image matters less to RetinaNet.

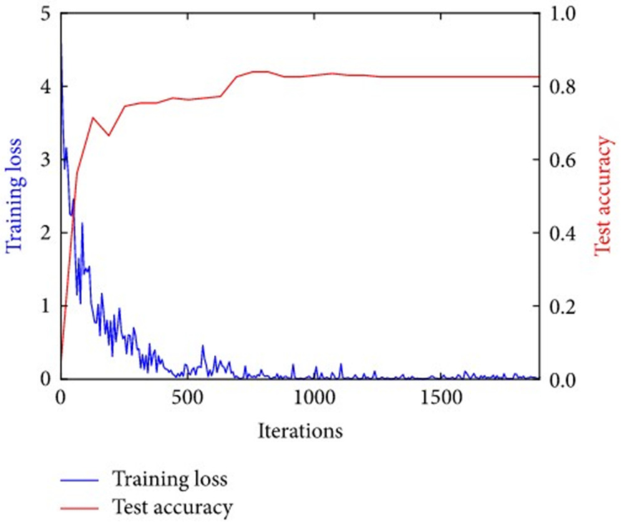

Fig. 1. R-CNN performance.

As shown in Figure 1, there is a considerable increase in the testing accuracy whereas there is a decrease in the training loss for the first 1000 iterations. Similarly, as shown in Figure 2, when the iterations increase, the training loss decreases linearly and the test accuracy increases significantly. For the first 600 iterations the training loss is close to 0.3. As shown in Figure 3, with an increase in the training iterations, the loss decreases linearly and the test accuracy increases significantly for the first 3000 iterations, whereas the training loss is close to 0 after that. Table II shows the comparison of the results of the current and previous research. We have outperformed the Faster R-CNN model of previous research results by 12% in mAP. We achieved high accuracy by training the dataset on the RCNN model, reaching 99.8%, a difference of 8.8% from [6]. Our model also achieved a high accuracy of 95.2% with RetinaNet model with threshold 0.5, a difference of 0.6% behind the model presented in [24], at the same threshold.

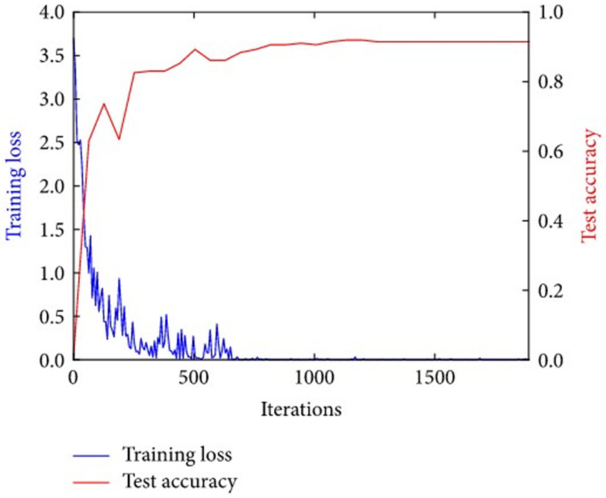

Fig. 2. Faster-RCNN performance.

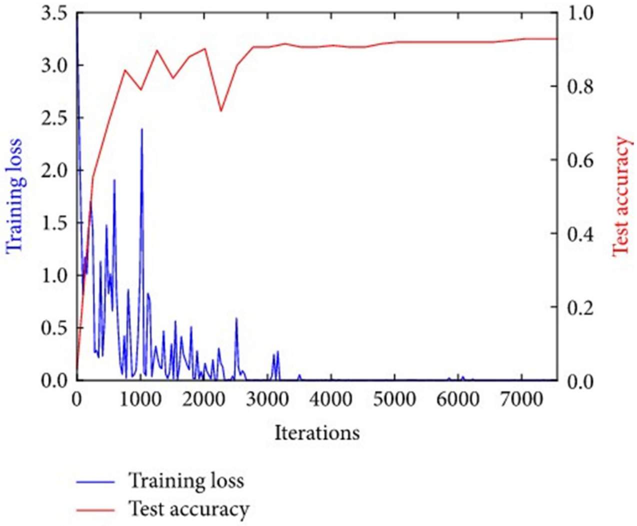

Fig. 3. RetinaNet performance.

TABLE II. RESULT COMPARISON FOR METHODS THAT USED THE FLICKRLOGOS-32 DATASET

| Method | Precision | Recall | Accuracy | mAP |
|---|---|---|---|---|
| [25] | 0.909 | 0.845 | 0.884 | - |
| [26] | - | - | 0.896 | - |
| [2] | 0.955 | 0.908 | - | - |
| [6] | 0.976 | 0.676 | 0.910 | - |
| [18] | - | - | 0.928 | - |
| [12] | - | - | - | 0.464 |
| [15] | - | 0.798 | - | 0.565 |
| [22] RetinaNet: threshold 0.5 | 0.948 | 0.829 | 0.946 | 0.620 |
| Ours: RetinaNet: threshold 0.5 | - | - | 0.952 | 0.40 |
| Ours: RCNN | - | - | 0.998 | 0.65 |
| Ours: Faster R-CNN | - | - | 0.936 | 0.74 |

## VI. CONCLUSION

Applying CNNs in LD is a common process for such purposes. Many network architectures have been applied, resulting in varying accuracy. The process involves many challenges as the logos may appear on any scale, position, and under different perspectives in an image. The traditional techniques for LR include key point-based sensors and local feature-based recognition. However, by using different CNN models such as R-CNN and FR-CNN, better results can be achieved as it has been discussed in this study. As newer CNN models get available, more experiments should be done, taking into account the trade-offs between accuracy, training and development cost.

## REFERENCES

[1] M M. Bastan, H.-Y. Wu, T. Cao, B. Kota, and M. Tek, “Large Scale Open-Set Deep Logo Detection,” Nov. 2019, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1911.07440.

[2] G. Oliveira, X. Frazao, A. Pimentel, and B. Ribeiro, “Automatic graphic logo detection via Fast Region-based Convolutional Networks,” in *International Joint Conference on Neural Networks*, Vancouver, Canada, Jul. 2016, pp. 985–991, https://doi.org/10.1109/IJCNN.2016.7727305.

[3] P. Chakraborty and C. Tharini, “Pneumonia and Eye Disease Detection using Convolutional Neural Networks,” *Engineering, Technology & Applied Science Research*, vol. 10, no. 3, pp. 5769–5774, Jun. 2020, https://doi.org/10.48084/etasr.3503.

[4] M. Salemdeeb and S. Erturk, “Multi-national and Multi-language License Plate Detection using Convolutional Neural Networks,” *Engineering, Technology & Applied Science Research*, vol. 10, no. 4, pp. 5979–5985, Aug. 2020, https://doi.org/10.48084/etasr.3573.

[5] S. C. H. Hoi *et al.*, “LOGO-Net: Large-scale Deep Logo Detection and Brand Recognition with Deep Region-based Convolutional Networks,” Nov. 2015, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1511.02462.

[6] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, “Deep learning for logo recognition,” *Neurocomputing*, vol. 245, pp. 23–30, Jul. 2017, https://doi.org/10.1016/j.neucom.2017.03.051.

[7] S. Yang, J. Zhang, C. Bo, M. Wang, and L. Chen, “Fast vehicle logo detection in complex scenes,” *Optics & Laser Technology*, vol. 110, pp. 196–201, Feb. 2019, https://doi.org/10.1016/j.optlastec.2018.08.007.

[8] C. Eggert, D. Zecha, S. Brehm, and R. Lienhart, “Improving small object proposals for company logo detection,” in *ACM on International Conference on Multimedia Retrieval*, New York, USA, Jun. 2017, pp. 167–174.

[9] H H. Jung *et al.*, “Detection of masses in mammograms using a one-stage object detector based on a deep convolutional neural network,” *PLoS ONE*, vol. 13, no. 9, Sep. 2018, Art. no. e0203355.

[10] C. Eggert, S. Brehm, A. Winschel, D. Zecha, and R. Lienhart, “A closer look: Small object detection in faster R-CNN,” in *IEEE International Conference on Multimedia and Expo*, Hong Kong, China, Jul. 2017, pp. 421–426, https://doi.org/10.1109/ICME.2017.8019550.

[11] H. Su, S. Gong, and X. Zhu, “Scalable logo detection by self co-learning,” *Pattern Recognition*, vol. 97, Jan. 2020, Art. no. 107006, https://doi.org/10.1016/j.patcog.2019.107003.

[12] A. Tuzko, C. Herrmann, D. Manger, and J. Beyerer, “Open Set Logo Detection and Retrieval,” Oct. 2017, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1710.10891.

[13] H. Su, S. Gong, and X. Zhu, “WebLogo-2M: Scalable Logo Detection by Deep Learning from the Web,” in *IEEE International Conference on Computer Vision Workshops*, Venice, Italy, Oct. 2017, pp. 270–279, https://doi.org/10.1109/ICCVW.2017.41.

[14] G. Zhu and D. Doermann, “Automatic Document Logo Detection,” in *Ninth International Conference on Document Analysis and Recognition*,

Parana, Brazil, Sep. 2007, vol. 2, pp. 864–868, https://doi.org/10.1109/ICDAR.2007.4377038.

[15] I. Fehervari and S. Appalaraju, "Scalable Logo Recognition Using Proxies," in *IEEE Winter Conference on Applications of Computer Vision*, Waikoloa Village, USA, Jan. 2019, pp. 715–725, https://doi.org/10.1109/WACV.2019.00081.

[16] P. F. Jaeger *et al.*, "Retina U-Net: Embarrassingly Simple Exploitation of Segmentation Supervision for Medical Object Detection," vol. 1811, Nov. 2018, Accessed: Jan. 08, 2021. [Online]. Available: http://adsabs.harvard.edu/abs/2018arXiv181108661J.

[17] H. Su, X. Zhu, and S. Gong, "Deep Learning Logo Detection with Data Expansion by Synthesising Context," in *IEEE Winter Conference on Applications of Computer Vision*, Santa Rosa, USA, Mar. 2017, pp. 530–539, https://doi.org/10.1109/WACV.2017.65.

[18] A. Alsheikhy, Y. Said, and M. Barr, "Logo Recognition with the Use of Deep Convolutional Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 10, no. 5, pp. 6191–6194, Oct. 2020, https://doi.org/10.48084/etasr.3734.

[19] 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*, Apr. 2011, Art. no. 25, https://doi.org/10.1145/1991996.1992021.

[20] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *IEEE International Conference on Computer Vision*, Venice, Italy, Oct. 2017, pp. 2999–3007, https://doi.org/10.1109/ICCV.2017.324.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Oct. 2014, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1311.2524.

[22] S S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, https://doi.org/10.1109/TPAMI.2016.2577031.

[23] F F. Al-Azzo, A. M. Taqi, and M. Milanova, "Human Related-Health Actions Detection using Android Camera based on TensorFlow Object Detection API," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, pp. 9–23, 2018.

[24] F. S. Herrera and J. M. Saavedra, "DLDENet: Deep Local Directional Embeddings with Increased Foreground Focal Loss for object detection," in *38th International Conference of the Chilean Computer Science Society*, Concepcion, Chile, Nov. 2019, pp. 1–8, https://doi.org/10.1109/SCCC49216.2019.8966436.

[25] S. Bianco, M. Buzzelli, D. Mazzini, and R. Schettini, "Logo Recognition Using CNN Features," in *Image Analysis and Processing — ICIAP 2015*, V. Murino and E. Puppo, Eds. New York, USA: Springer, 2015, pp. 438–448.

[26] F. N. Iandola, A. Shen, P. Gao, and K. Keutzer, "DeepLogo: Hitting Logo Recognition with the Deep Neural Network Hammer," Oct. 2015, Accessed: Jan. 08, 2021. [Online]. Available: http://arxiv.org/abs/1510.02131.