



Article

LogoNet: A Robust Layer-Aggregated Dual-Attention Anchorfree Logo Detection Framework with an Adversarial Domain Adaptation Approach

Rahul Kumar Jain ¹, Taro Watasue ², Tomohiro Nakagawa ², Takahiro Sato ², Yutaro Iwamoto ¹, Xiang Ruan ^{2,*} and Yen-Wei Chen ^{1,*}

¹ Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga 525-8577, Japan; rahulkumarjain16@gmail.com (R.K.J.); yiwamoto@fc.ritsumei.ac.jp (Y.I.)

² tiwaki Co., Ltd., Kustasu, Shiga 525-8577, Japan; watasue@tiwaki.com (T.W.); nakagawa@tiwaki.com (T.N.); sato@tiwaki.com (T.S.)

* Correspondence: ruanxiang@tiwaki.com (X.R.); chen@is.ritsumei.ac.jp (Y.-W.C.)

Abstract: The task of logo detection is desirable and important for various fields. However, it is challenging and difficult to identify logos in complex scenarios as a logo can appear in different styles and platforms. Logo images include diverse contexts, sizes, projective transformation, resolution, illumination and fonts, which make it more difficult to detect a logo. To address these issues, we presented a deep learning-based algorithm for logo detection called LogoNet. It includes an hourglass like top-down bottom-up feature extraction network, a spatial attention module and an anchorfree detection head similar to CenterNet. In order to improve performance, in this paper, an extended version of LogoNet is proposed, called—Dual-Attention LogoNet, that exploits different attention mechanisms more efficiently. The incorporated channel-wise and spatial attention modules refine and generate robust and balanced feature maps to predict visual and semantic information more accurately. In addition, we propose a lightweight architecture for both LogoNet and Dual-Attention LogoNet for practical applications. The proposed lightweight architecture significantly reduces the number of network parameters and improves the inference time to address the real-time performance while maintaining accuracy. Furthermore, to address the domain shift problem in practical applications, we also propose an adversarial-learning-based domain adaptation approach, which is easily adaptable to any anchorfree detectors. Our attention-based method shows a 1.8% improvement in accuracy compared to the state-of-the-art detection network on the FlickrLogos-32 dataset. Our proposed domain adaptation approach significantly improves performance by 1.3% mAP compared to direct transfer on the target domain without increasing any labeling cost and network parameters.

Keywords: deep learning; anchorfree; HourglassNet; attention mechanism; lightweight CNNs; CenterNet



Citation: Jain, R.K.; Watasue, T.; Nakagawa, T.; Sato, T.; Iwamoto, Y.; Ruan, X.; Chen, Y.-W. LogoNet: A Robust Layer-Aggregated Dual-Attention Anchorfree Logo Detection Framework with an Adversarial Domain Adaptation Approach. *Appl. Sci.* **2021**, *11*, 9622. <https://doi.org/10.3390/app11209622>

Academic Editors: Jose Santamaria and Zong Woo Geem

Received: 19 September 2021

Accepted: 13 October 2021

Published: 15 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Logo detection has now become a demanding task as it is applicable in many applications such as brand promotion, social media monitoring, intelligent transportation, auto-driving, illegal/fraud logo detection and market research. Logo detection is also very useful for analyzing and tracking advertisements on different platforms. However, detection of logos in real-world images is a difficult task because there are countless brands in the world and logos of each brand may have diverse context, projective transformation, resolution and illumination. Logos may have unknown fonts, different sizes and colors on diverse platforms. In real scenarios the logo appears as a small object entity compared to the resolution of the images in which it presents. Moreover, inter-class similarity and intra-class difference in the logo images make the logo detection task even more difficult [1].

Since the evaluation of convolution neural networks, deep learning-based detectors have become the leading framework for object detection [2–4]. Several object detection methods have been proposed in the last decade, from two-stage region proposal-based Faster R-CNN [5] to anchor-based methods such as YOLO [6] and SSD [7]. Since then, object detection methods based on deep learning have been used in logo detection.

István et al. [1] trained a Faster R-CNN model [5] to classify logo and non-logo objects in a class-agnostic manner, they trained a separate network [8] to retrieve logo images. Su et al. [9,10] proposed to use data augmentation to create synthesized logo images for model learning. Su et al. [11] presented the model self-learning principle using logo images collected on the web. They trained a model iteratively and identified the most compatible logo images from a noisy dataset. These selected images are then used to learn the model. In [12], the authors presented model self-co-learning method with the last method. They trained two different detectors [5,6] to identify compatible training logo images from the noisy dataset. These identified training images have been fed as an input in a cross-model manner. Jain et al. [13] proposed a weakly supervised logo detection algorithm by implementing dual-attention based mechanism with the DRN network to recognition logo without using bounding box annotated training data. Although training images are synthetically or automatically generated, the results do not show satisfactory performance on real images.

Fine-tuning of these detectors usually requires tuning of various hyperparameters like size, number and aspect ratio of densely placed anchor boxes. These methods require careful design for RoIs, sizes and number of anchor-boxes. Their experimental studies show that the accuracy of two-stage detectors such as the Faster R-CNN is better than that of one-stage anchor-based detectors such as SSD, but expensive in terms of resources and detection speed. On the other hand, one-stage detectors have shown fast inference time but the accuracy is not complete in some cases. In most real applications, logo detection tasks operate at a low spec. Devices such as mobile phones or IP cameras that require algorithms to be both lightweight and have high accuracy. For a better trade-off between accuracy and computational speed, here, we present an attention-based feature extraction network with an anchor-free detector [14], called Dual-Attention LogoNet. This paper is an extension version of our ICCE-2021 conference paper [15]. Here, we add a channel-wise attention module together with spatial attention module to generate more balanced feature maps. Our goal is to focus on improving accuracy with attention mechanisms and to build a lightweight model which is more feasible to deploy on embedded edge computing devices. Here, we also present a novel adversarial domain adaptation-based method for practical logo detection.

Recently, various anchor-free detection methods have been designed by researchers for detection task. These anchor-free detection methods are capable of achieving better performance than the abovementioned detection methods. These methods overcome the problem of class-imbalance of RoIs proposals and the critical anchor-box design choice by locating objects in terms of keypoints. Law et al. [16] proposed CornerNet for detecting objects as a pair of corners of a bounding box. The method was later improved by Duan et al. [17], in which authors proposed to detect objects as center, top-left and bottom-right points. ExtremeNet is presented by Zhou et al. [18]. ExtremeNet detects objects by identifying a single center point and four extreme points in different directions. Zhou et al. [14] also proposed a method to detect object by its center point, they therefore named its algorithm CenterNet.

In recent years, attention architecture has become popular in deep learning tasks, which is also used by many new proposed object detection algorithms. Such methods have proved to be useful for refining and emphasizing informative features. Wang et al. [19] proposed a method to enhance the spatial features using the mask module. This module is employed with a trunk branch consisting of bottom-up and top-down feedforward structure. Hu et al. [8] introduced SENet module for calculating channel-wise weights of a convolutional layer to capture channel-wise responses. Wang et al. [20] proposed ECANet

block to model channel-wise features more effectively and efficiently. Chen et al. [21] proposed an attention mechanism network to classify and localize liver lesions on CT images. Woo et al. [22] proposed to use channel and spatial attention blocks within the convolutional block. Zhu et al. [23] proposed a network for learning spatial information using the attention mechanism. They added the calculated attention weights to the output of the classification layer.

Normally, training of deep learning-based models follows a supervised learning scheme and relies on large annotated training datasets. A deep learning model suffers performance degradation due to domain shift (source-to-target domain) during inference time [24,25]. In practice, such performance degradation limits the scalability and applicability of deep learning-based models. On the other hand, fine-tuning a model on new domain might face the problem of lack of training data because object-level annotation is basically a time-consuming and labor-intensive task. Training a well-generalized model which is able to be applied to different domains is a hot research topic today. As a result, recently, several domain adaptation-based methods have been proposed to learn model from one domain and generalize well to another domain [24–27].

Inspired by the existing adversarial learning-based domain adaptation method [25,26] which has been developed primarily for segmentation applications, in this work, we present a domain adaptation method for logo detection using adversarial learning to mitigate errors caused by domain shift. We have used annotated logo images (source-domain) and unlabeled logo images (target-domain) for training to bring closer these source and target domains. The added discriminator-based network can be learned into an end-to-end manner like a normal detector. Since anchor-free detectors train the network to learn objects in terms of some keypoints, we propose to use mid-level output feature maps instead of class-wise heatmaps to align the distribution of target and source domains. Our adversarial learning approach is motivated by the fact that the use of mid-level outputs benefits from robust information about the domain while retaining object-level information. This method can be easily adapted to other anchor-free detectors.

2. Proposed Network

Our architecture includes a feature extractor backbone, spatial and channel attention modules and a detection head. Inspired by HourglassNet [18] we use a top-down bottom-up network. However, different from conventional network, we aggregate both convolutional layer output feature maps within each residual block. A skip layer connection is added with this output and provided as input for the next convolution block. In our proposed method the final feature maps is generated by combining the outputs obtained by two stacked hourglass networks. To precisely emphasize the attributes of target objects in the generated feature maps, we employ a channel-wise attention module along with the spatial attention module after the feature extractor network. What makes our architecture better in detecting logos than conventional detectors is the newly added two attention modules prior to the detection head. The two branches using channel and spatial attention modules, respectively, produce category-wise keypoint heatmaps of the input images. Such feature maps are generated by their respective attention modules to emphasize the network capacity of learning longer-range dependencies and help to know what and where can be found in the image. For accurate detection of target logos in feature maps, we perform matrix element-wise addition to these two category-wise feature maps. The aggregated final feature maps is given as input to detection head. The detection head is similar to CenterNet. The overall architecture of LogoNet is shown in Figure 1.

The detail of architecture is described as follows. Section 2.1 provides detail about feature extractor network. The spatial attention module and channel-wise attention module are explained in Sections 2.2 and 2.3, respectively. Detection head is described in Section 2.4. Lightweight-CNNs models are reported in Section 2.5. The Domain-Adaptation-based logo detection method is described in Section 2.6.

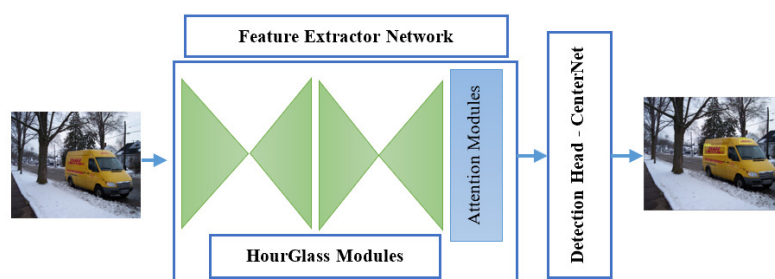


Figure 1. Overall network architecture of Dual-Attention LogoNet.

2.1. Feature Extractor

Hourglass network was introduced for the human pose estimation task by Newell et al. [28]. The network consists of bottom-up and top-down structured modules, where input channels are expanded and dimensions of the feature maps are down-sampled by a series of convolutional, stride and max-pooling operations. Subsequently, upsampling operations are performed to produce symmetric feature map blocks in hourglass style. Skip connections are added during upsampling to prevent the loss of information. Hourglass network was used in CornerNet [16] for object detection. After that, the same framework was used in CenterNet [14]. Our hourglass-like feature extractor network employs the same arrangement of convolution blocks as [16].

In the feature extractor network, first, input feature maps ($3 \times 128 \times 128$) are passed through a convolutional block which reduces the input dimension by half by using a 7×7 convolutional operations and a stride of size 2 with 128 channels. After that the feature maps are fed into a residual block with 3×3 convolutional operations and a stride of size 2. It produces a feature maps with 256 channels and spatial dimension of 128×128 . Subsequently, feature maps are fed into stacked hourglass modules to produce feature maps with global spatial and semantic information. Hourglass module consists of bottom-up and top-down design with residual learning blocks. There are five stages in downsampling and upsampling operations. The processing modules at each stage, including the skip connection modules (there are skip connections between symmetric blocks of a hourglass module, referring to Figure 2), consists of two residual blocks. Each residual block includes two convolutional layers and one skip connection layer. The spatial dimension of the feature map is reduced by a stride of size 2 which is employed for the first convolutional operation in the residual block. The rest of the convolution operations (including the second residual block) use a stride of size 1 and keep the spatial dimension unchanged. The kernel size of 3×3 is used in every convolutional operation. The skip connection layer in the residual blocks uses linear transformation (1×1 convolution) and matches the spatial and channel dimension of the input feature maps with the output of the convolution layer. The spatial resolution of feature maps is reduced by 5 times and the number of channels increases as [256, 384, 384, 384, 512] along the way. Upsampling of feature maps is performed using the nearest-neighbor algorithm, followed by two residual blocks at each stage. The final output feature map has 256 channels and a 128×128 spatial dimension. The detailed structure of the hourglass module is described in Table 1.

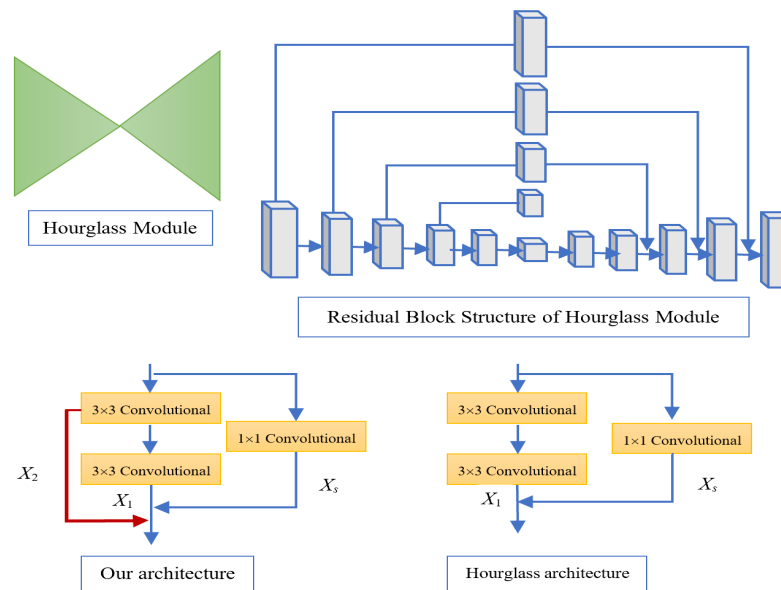


Figure 2. Illustration of proposed aggregation of various layers in a convolution block.

Table 1. The detailed operation and parameters of each layer in an hourglass module.

Layer Name	Output Dimension	Operation, Kernel Size, Output Channels, Stride	Layer Name	Output Dimension	Operation, Kernel Size, Output Channels, Stride
Conv1_1	64×64	Conv, 3×3 , 256, 2	Conv10_2	128×128	Conv, 3×3 , 256, 1
Conv1_2	64×64	Conv, 3×3 , 256, 1	Conv10_1	128×128	Conv, 3×3 , 256, 1
Conv2_1	32×32	Conv, 3×3 , 384, 2	Conv9_2	64×64	upsampling
Conv2_2	32×32	Conv, 3×3 , 384, 1	Conv9_1	64×64	Conv, 3×3 , 384, 1
Conv3_1	16×16	Conv, 3×3 , 384, 2	Conv8_2	32×32	upsampling
Conv3_2	16×16	Conv, 3×3 , 384, 1	Conv8_1	32×32	Conv, 3×3 , 384, 1
Conv4_1	8×8	Conv, 3×3 , 384, 2	Conv7_2	16×16	upsampling
Conv4_2	8×8	Conv, 3×3 , 384, 1	Conv7_1	16×16	Conv, 3×3 , 384, 1
Conv5_1	4×4	Conv, 3×3 , 512, 2	Conv6_2	8×8	upsampling
Conv5_2	4×4	Conv, 3×3 , 512, 1	Conv6_1	8×8	Conv, 3×3 , 512, 1
					upsampling

Based on the original hourglass architecture, our proposed network densely aggregates convolutional layers into each residual block at different scales. Each residual block has two convolutional layers and a skip connection layer. Residual learning uses a skip connection to add with the output of the second convolutional layer. We propose to aggregate outputs of both convolutional layers with skip connection within each convolutional block inspired by [29]. In each residual block, both convolutional operations and the skip connection layer generate feature maps of the same spatial and channel dimensions so that these output feature maps can be directly added without increasing network overhead.

$$X = X_s + X_1 + X_2 \quad (1)$$

where the input feature map passes through convolutional operations, X_1 and X_2 are the output of the two convolutional operations. X_s denotes output feature maps of the

skip connection layer. Figure 2 illustrates the residual block structures of the hourglass network [28] and our proposed approach.

In order to project important information, we added the output feature maps of both stacked hourglass modules. This final output is provided to the attached attention modules. This avoids the loss of information and detail during the downsampling and upsampling operation of feature maps. Our experiments show that our approach generates a robust feature map without raising any computation cost. Figure 3 illustrates the overall framework of CenterNet, LogoNet and Dual-Attention LogoNet.

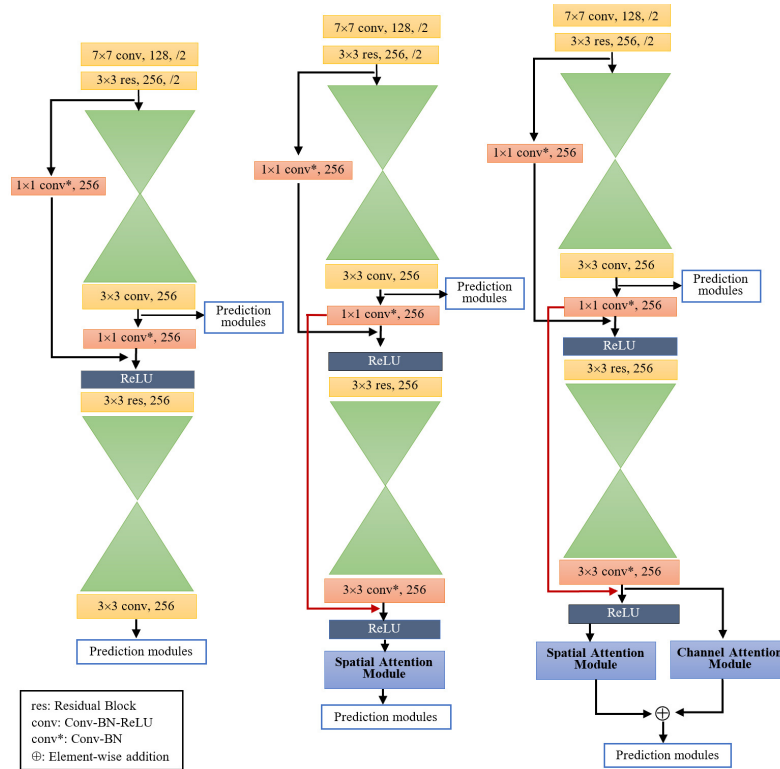


Figure 3. (Left) CenterNet framework. (Middle) LogoNet with spatial attention module and added final output feature maps. (Right) Dual-Attention LogoNet.

2.2. Spatial Attention Module

We produce spatial attention weights using the inter-spatial relationships of channels to obtain rich and global spatial information that helps to create a robust global feature map. Figure 4 depicts the overview of our proposed spatial attention module. A feature map $A \in R^{C \times H \times W}$ is provided as an input to the spatial attention module where C denotes channel size and $H \times W$ are height and width of the feature map, which are $256 \times 128 \times 128$ in this paper. This input A is then fed into a 1×1 linear transformation layer and a normalized feature map $S_{sigmoid}$ is created for all channels using the sigmoid activation function.

$$S'_{ij} = \frac{1}{1 + \exp(-S_{ij})} \quad (2)$$

where S_{ij} is the scalar value at i th and j th position and S'_{ij} denotes corresponding activated scalar value at i th and j th pixel position. The output of this operation is a sigmoid activated map, i.e., $S_{sigmoid} \in R^{C \times H \times W}$.

Additionally, the input $A \in R^{C \times H \times W}$ is fed into a convolutional block, which generates a feature map (F_{CONV3}). This convolutional block consists of three convolutional layers with $1 \times 1, 3 \times 3, 1 \times 1$ kernel size, respectively. To keep channel-wise details, the number of channels (C) for each convolutional layer remains unchanged which is 256. ReLU activation is followed by the first two convolutional operations while batch normalization has been

performed for all the convolutional layers. Softmax normalization strategy is applied across the channels over the output feature space of the convolutional block (F_{CONV3}). During softmax normalization, all positional scalar values in the same pixel-position across all feature channels are considered. New scalar value is synthesized for each pixel across the channels using the value of other pixels at the same index. In Equation (3), if $P_{i,j,k}$ is a scalar value at i th and j th pixel position in k th channel, a normalized scalar value $P'_{i,j,k}$ can be obtained as:

$$P'_{i,j,k} = \frac{\exp(P_{i,j,k})}{\sum_{k=1}^C \exp(P_{i,j,k})} \quad (3)$$

where C denotes the number of channels in feature map F_{CONV3} . A softmax normalized feature maps $P_{softmax} \in R^{C \times H \times W}$ has been produced using these normalized scalar values ($P'_{i,j,k}$).

We perform element-wise product of both generated normalized feature map, i.e., $S_{sigmoid}$ and $P_{softmax}$. The input feature map (A) is added as a skip connection to this product to obtain final attention-weighted feature map.

$$A_{attention} = A + (S_{sigmoid} \odot P_{softmax}) \quad (4)$$

where \odot is the element-wise product.

As we mentioned in our previous conference paper [15], our convolutional layers block follows the module structure proposed in [23], but our method is totally different from their approach. For multi-label image classification, they employed a regularization module to generate attention weights. These attention weights were provided to the classification layer of the feature extractor network which was ResNet [30]. Whereas, we generate a weighted feature map and perform an element-wise addition with the input to obtain a robust representation of input image. Our proposed technique uses both sigmoid and softmax functions as activation to learn important spatial weights.

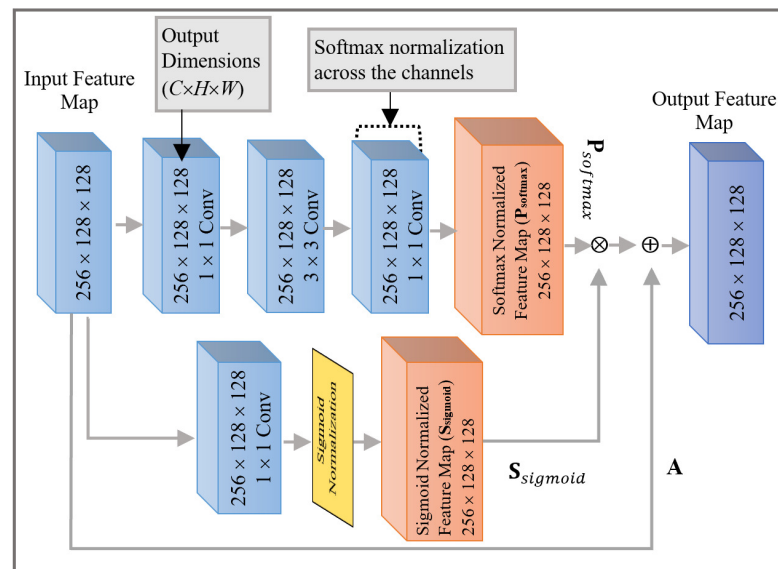


Figure 4. Illustration of the proposed spatial-attention module.

2.3. Channel-Wise Attention Module

To capture channel-wise attention weights, Wang et al. [20] introduced ECANet block. To achieve channel-wise dependencies, global-average pooling (GAP) is performed on the input feature maps. Subsequently a 1-D convolutional operation is employed to learn cross-channel interaction. A sigmoid activation function operates at this layer to learn channel-wise attention weights. They proposed to use an adaptive kernel size to capture

local cross-channel interactions by considering a channel and its k neighbors (coverage of interaction). In their method the kernel size k is proportional to the number of channels. Channel-wise response is emphasized by multiplying the attention weights with the input feature maps. This weight-enhanced feature maps is added to the input feature map as the final output.

In our proposed method, ECANet [20] module with a kernel size of 3 is used. Unlike the proposed approach, we directly use the attention-based feature maps to produce category-wise heatmaps without adding the input feature maps as skip connection. Figure 5 shows the channel attention module.

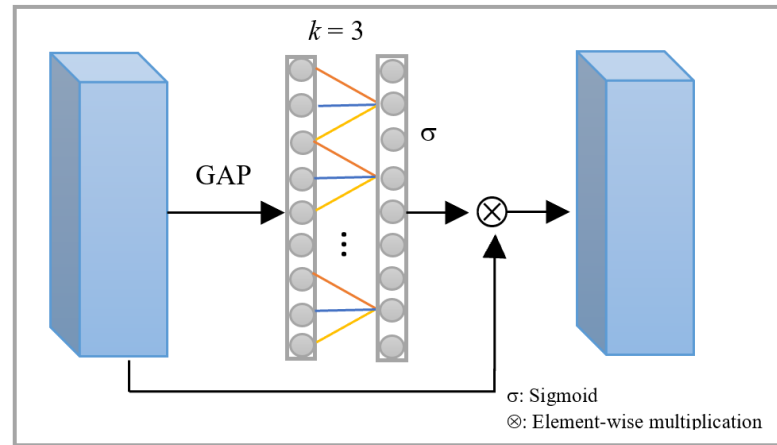


Figure 5. An overview of channel-wise attention module.

2.4. Detection Head—CenterNet

CenterNet is an anchor-free detector proposed in [14]. CenterNet identifies objects as a point at their bounding box center. During training, CenterNet converts ground truth RoIs into heatmaps. For the training image a keypoint map $K_{(x,y,c)}$ is generated in which if the coordinates (x,y) belong to the center of an object then it will be activated and the rest positions will be set to zero. The keypoint map is then converted into a corresponding set of heatmaps. These heatmaps are used to train the detector with a focal loss function to classify into corresponding class (L_k) [31]. CenterNet also consists of an offset head for object location and a size head to regress the size of object to generate its RoIs. The final detection loss function is given as:

$$L_{det} = L_k + \lambda_{size}L_s + \lambda_{off}L_{off} \quad (5)$$

where L_s and L_{off} are L1 loss functions and λ_{size} and λ_{off} are loss weights.

L1 loss or L1 regularization is used to calculate the error, where error is the difference between the ground truth bounding box and the predicted bounding box coordinates. During detection, class-wise heatmaps are generated corresponding to separate categories. Then some peak points are found out in the generated class-wise heatmaps. In the normal setting, 100 peak points are considered for detection within each category. A keypoint estimator $\hat{\gamma}$ is used to predict all center points. A set of n detected center point \hat{P}_c for all c classes is estimated as $\hat{P} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^n$ where (x_i, y_i) is the integer coordinate for a keypoint location. Detection confidence score is measured using the keypoint values $\hat{Y}_{x_i, y_i, c}$. A local offset is also predicted for center point location and to regress object size. For the learned model, evaluation metric in terms of mean average precision (mAP) is determined for all classes when the Intersection over Union (IoU) with the ground truth bounding box exceeds 0.5. The precision value for any given category is the percentage of correct predictions. i.e.,

$$Precision = TruePositive / (TruePositive + FalsePositive) \quad (6)$$

Whereas recall measures the proportion of true positive that can be determined as:

$$Recall = TruePositive / (TruePositive + FalseNegative) \quad (7)$$

In general, the average precision (AR) for any given category is the area under the precision-recall curve (AUC—area under the curve). The mean average precision (mAP) is the average value of the AR for all categories of a dataset.

In this study, we present an architecture containing spatial and channel attention modules as an extension of the our previous method. A conventional way of implementing the channel attention module is that attention blocks are added to each convolutional block during feature extraction [8,20]. While some methods proposed to use both spatial and channel attention mechanisms within each convolutional block [22]. In our proposed method we employ both attention modules only once in parallel order just before the generation of the class-wise heatmaps, which are used to make dense predictions. The spatial attention and channel attention modules generate two sets of class-wise heatmaps. This arrangement captures strong informative spatial features along with high-level semantics features. Element-wise addition of class-wise feature maps, generated by both attention modules, is performed for better fusion of class-wise information.

2.5. Lightweight Model

To build a compact network and improve the detection speed for practical applications, we present a Lightweight architecture. We embed factorization of standard convolutions inspired by MobileNetv2 [32]. In our lightweight module, a convolutional operation comprises a combination of pointwise and depthwise separable convolutional layer. Pointwise is a standard 1×1 convolution operation that performs linear transformation of the input and changes the channel dimensionality. Depthwise convolution applies a single filter per each channel to filter the features. Our network uses Batchnorm and ReLU activation operation after the depthwise convolutional layer. The same pattern of layers is followed for skip connection layers. Spatial dimension is handled by the max-pooling operation. This design is used with the LogoNet and Dual-Attention LogoNet architecture. We convert each standard residual convolution block of our architecture into a depthwise convolution block that follows the approach of MobileNetv2 block. We employ the approach only for hourglass module layers, feature transformations of other layers including the attention modules is performed using a standard convolution operation. This approach reduces network complexity and computation compared to standard convolution. Depthwise computation can be expressed as:

$$\hat{O}_{l,m,c} = \sum_{i,j} \hat{K}_{i,j,c} \cdot F_{l+i-1,m+j-1,c} \quad (8)$$

where F and \hat{O} are input and output feature maps with C number of channels. \hat{K} is a depthwise convolution kernel of size $D_K \times D_K \times D_C$ where D_K is the size of kernel, which is 3 in our case. For a feature map of D_F height and width, the total computation cost of depthwise and pointwise convolution operation can be computed as:

$$C_{in} \cdot C_{out} \cdot D_F \cdot D_F + D_K \cdot D_K \cdot C_{out} \cdot D_F \cdot D_F \quad (9)$$

exploits where C_{in} and C_{out} are the input and output channels.

To compare the proposed architecture, we also demonstrate lightweight models, exploring the CP-Decomposition (CPD) [33]. The CPD method is the typical method for reducing complexity, which factorizes a tensor into a sum of outer products of vectors. For a given tensor of 3-dimensional space, the CP decomposition can be explained as:

$$T \approx \sum_{r=1}^R l_r \circ m_r \circ n_r \quad (10)$$

where $R > 0$, and l_r, m_r, n_r are vectors of relevant dimension, and ‘ \circ ’ denotes the outer product of two tensors, i.e.,

$$t_{i,j,k} \approx \sum_{r=1}^R l_{ri} \circ m_{rj} \circ n_{rk} \quad (11)$$

In case of rank one assumption of CPD (i.e., $R = 1$), the 4D kernel $\hat{C} \in R^{X \times Y \times Z \times S}$ will be separated into cross-products of four 1D filters as follows:

$$\hat{C} = \alpha \times \beta \times \gamma \times \eta \quad (12)$$

where α, β, γ are 1D convolution vectors convolving across the dimensions and the fourth corresponds to channels.

Here, we converted a standard convolution to two 1D convolutions within each residual block of proposed feature extractor. We use 1D convolution from two axes ($X \times 1$, $1 \times Y$) to convolve the feature maps. First, we convolve the features using single filter each channel (depthwise) by a kernel size of (3×1) . Then a kernel of size (1×3) is applied to map the number of feature channels. Same approach is applied with skip connection layer to transform the feature maps. Block structures of feature extractor with depthwise convolution and CPD methods are shown in Figure 6.

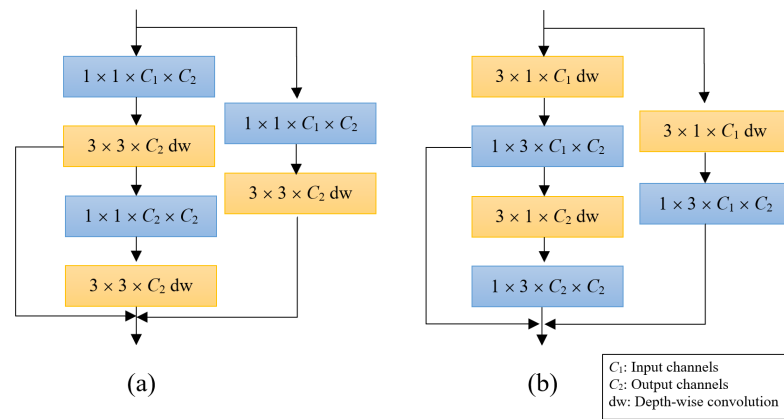


Figure 6. (a) Convolutional block with our lightweight module (b) Convolutional block with CPD method.

2.6. Adversarial-Based Domain Adaptation for Anchor-Free Detector

In practical applications, we need to apply the trained model (i.e., LogoNet) to a new dataset (target dataset). The model always suffers performance degradation due to domain shift because the distribution of the source dataset (training dataset) is different from that of the target dataset (test dataset). To enhance the generalization of the model, we aim to address model learning towards the distribution of target domain by aligning the output feature maps of source and target domains as close to each other as possible. In order to align the model between two different domains, we exploit the adversarial learning scheme by adding a domain discriminator network in the training phase to detection framework. The architecture of the LogoNet framework with the proposed domain adaptation training scheme is shown in Figure 7, which consists of a feature extraction network and a detection module. The detection module has three heads, heatmaps-head (for generating class-wise heatmaps), offset-head (for identifying object locations), object-size head (for regressing the size of objects). The anchor-free detector generates class-wise heatmaps corresponding to each class using the output feature maps (mid-level output) of the feature extraction network. The offset and size output maps are also generated separately to give complete detection loss. Previously proposed adversarial learning-based schemes [24–27], which have been introduced primarily for semantic segmentation tasks, make use of the final class-wise output of the feature extraction network. Since anchor-free detectors train the

network to recognize objects in terms of some keypoints, we observed that the use of class-wise heatmaps leads to the loss of some important domain-specific information. It is very important to select the most suitable output feature maps to align the domain gap. In contrast to the previous methods, here we present a domain adaption-based LogoNet network, in which we propose to use the mid-level outputs of feature extraction network. The main advantage of using mid-level output is that it contains essential domain-specific semantic and visual information and is helpful to employ adversarial learning well. Using the design advantages of anchor-free detectors, we assume LogoNet generates mid-level output feature maps for images from the source domain and the target domain. The mid-level output maps of the source images rendered to different detection heads (heatmap-head, offset-head, size-head) to train the network for the respective tasks. Whereas, the mid-level output feature maps of the target images is used to calculate the adversarial loss to match the data distribution of source and target domains. Therefore, we do not need object-level annotations for the target images.

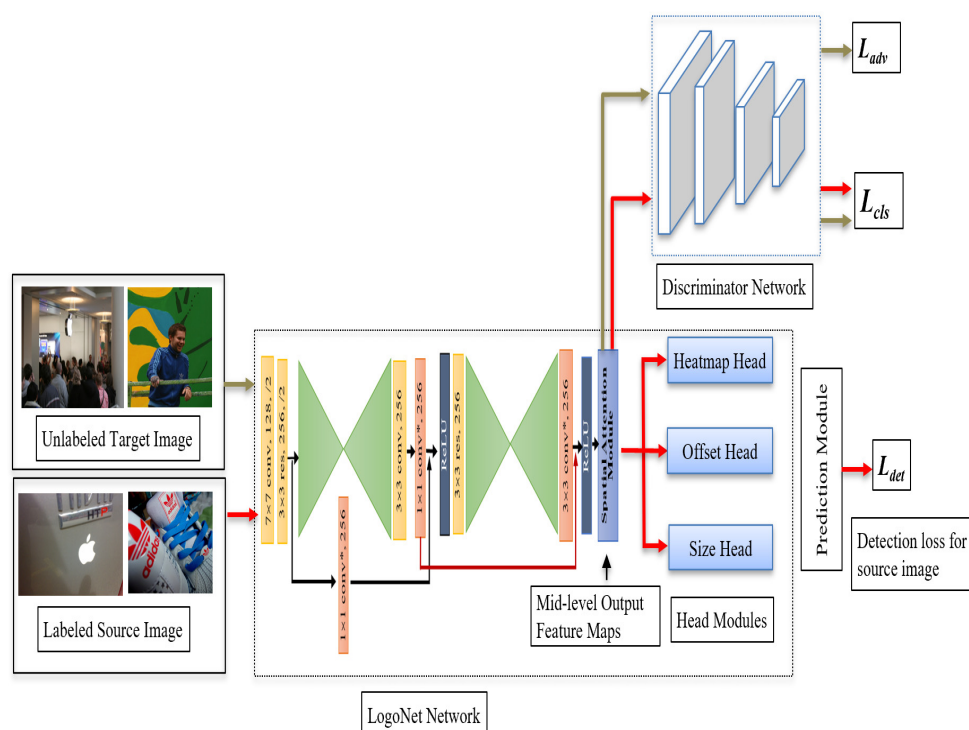


Figure 7. Network architecture of LogoNet with domain adaptation setting.

We assume that there are N images with corresponding object-level annotations in the source domain S with corresponding object-level annotations $\{x_i^s \in X_S, y_i^s \in Y_S\}$ where X_S is a set of input images in the source space, Y_S denotes the set of corresponding labels. Whereas, M is the number of images in the target domain T without object-level annotations $\{x_i^t \in X_T\}$, where X_T denotes the set of images in the target domain. To employ the adversarial learning technique, we add a domain discriminator network with the LogoNet framework that introduces the adversarial loss (L_{adv}) and classification loss (L_{cls}). The domain discriminator network consists of 5 convolution layers with a kernel size of 4×4 and a stride of size 2, each layer is coupled with a leaky-ReLU activation layer with a fixed negative slope of 0.2, except for the last convolution layer. The number of channels is [64, 128, 156, 512, 1] for each layer, respectively. Finally, a classification layer gives classification outputs. The detailed structure and operations of the discriminator network is described in Table 2.

Table 2. The design of the discriminator network.

Layer Name	Output Dimension	Operation, Kernel Size, output Channels, Stride
Layer1	128×128	Conv, 4×4 , 64, 2
Layer2	64×64	Conv, 4×4 , 128, 2
Layer3	32×32	Conv, 4×4 , 256, 2
Layer4	16×16	Conv, 4×4 , 512, 2
Layer5	4×4	Conv, 4×4 , 1, 2

We provide these mid-level outputs of the source image (Mid_X_S) and target image (Mid_X_T) as inputs to the discriminator network to classify the inputs from source domain (S) or target domain (T). The classification loss (L_{cls}) is calculated to update the network weights of the discriminator network to increase the ability to distinguish the inputs into the respective domains. We assign source images (source-domain) with domain label '0' and target images (target-domain) with domain label '1'.

The binary classification loss L_{cls} (training objective of domain discriminator network) can be defined as:

$$L_{cls} = \frac{1}{|X_S|} \sum_{i=1}^{|X_S|} L_{cls}(Mid_X_S^i, 0) + \frac{1}{|X_T|} \sum_{i=1}^{|X_T|} L_{cls}(Mid_X_T^i, 1) \quad (13)$$

where $Mid_X_S^i$ and $Mid_X_T^i$ are the mid-level features of the i th source training sample and the i th target training sample, respectively. $|X_S|$ and $|X_T|$ are sample numbers of source domain and target domain, respectively. Meanwhile, to bring the target domain (T) and source domain (S) distributions closer, we provide the mid-level output feature maps (Mid_X_T) of the target image into the discriminator network and compute the adversarial loss (L_{adv}) by giving an inverted domain label, i.e., '0' instead of '1'. The adversarial binary classification loss L_{adv} can be defined as:

$$L_{adv} = \frac{1}{|X_T|} \sum_{i=1}^{|X_T|} L_{cls}(Mid_X_T^i, 0) \quad (14)$$

Adversarial loss is propagated to update the gradients of LogoNet framework, the objective loss function of the network is given in the following equation.

$$L_{det} = L_k + \lambda_{size} L_s + \lambda_{off} L_{off} + \lambda_{adv} L_{adv} \quad (15)$$

λ_{adv} is loss weight. We use a value of 0.001 in our experiments. This approach encourages the network to produce similar output feature maps distributions from target (T) to the source domain (S) by mocking the discriminator network. The task-specific detection network and the domain discriminator network are jointly trained in an end-to-end manner. During inference we do not need the discriminator network and the normal detection pipeline is used to perform the detection task so we drop the discriminator network.

3. Experiments

3.1. Implementation

To evaluate the performance, we compare our proposed method with various methods such as CenterNet [14] (baseline), Faster R-CNN [5] and SSD [7]. The performance of the methods is measured in terms of mAP and detection time. For the CenterNet framework, training was conducted using a batch size of 2 for 140 epochs. We use HourglassNet-104 as feature extractor backbone pretrained on COCO dataset from ExtremeNet [18]. The initial learning rate is 1.25×10^{-4} which decreases by a multiplication of 0.1 at 90 and 120 epochs. The Adam optimizer is used for network optimization. A spatial resolution 512×512 is used for the input image. Faster R-CNN detector is trained with ResNet-50 backbone.

This model is trained for 50 epochs with batch size 4 and learning rate 0.001. SSD network is trained using VGG16 backbone with a batch size of 4 and initial learning rate of 0.001. The training is performed for 16,000 iterations. The experimental results are shown in percentage (%) of mAP value over all logo classes using Intersection of Union (IoU) value 0.5. Average inference time is given for one image. The inference time is calculated on our machine with Intel Core i7-8700 CPU, GeForce GTX 980 Ti GPU, Pytorch 0.4.1, CUDA 9.0 and CUDNN 7.1.

3.2. Evaluation on FlickrLogos-32 Dataset

Logo images of FlickrLogos-32 [34] dataset were used for training. FlickrLogos-32 dataset has 32 logo classes. Each class contains 70 images for experiments. For each class, we consider 30 images for training, 10 images for validation and 30 images for test. There were a total of 1602 logo objects in 960 test images for different categories.

Table 3 shows the details of ablation study on FlickrLogo-32 dataset. According to the results, the mAP accuracy is slightly improved when we aggregate feature maps at different scales (Proposed Method 1) or when we employ spatial attention module with baseline network (Proposed Method 2). When we implement spatial attention module with layer-aggregated feature maps together, detection accuracy improves effectively (LogoNet—Proposed Method 3). The calculation of the channel-wise response further improves the detection accuracy (Dual-Attention LogoNet—Proposed Method 4). We observe effectiveness of our methods in two steps: (i) the aggregation of feature maps at different scale, improves the global feature representation, (ii) combining attention modules with network generates a balanced and robust feature map with significant visual and semantic detail.

Table 3. Ablation Experiments on FlickrLogos-32 Dataset.

Methods	Layer-Aggregation	Spatial Attention	Channel Attention	mAP
CenterNet (baseline)				80.7
Proposed Method 1	✓			81.0
Proposed Method 2		✓		80.8
Proposed Method 3	✓	✓		82.2
Proposed Method 4	✓	✓	✓	82.5

Table 4 reports mAP and detection time using different detectors on Flickr32 dataset. These methods are: Faster R-CNN with ResNet50, SSD with VGG16, CenterNet with HourglassNet, CenterNet with SENet HourglassNet [8], CenterNet with ECANet HourglassNet [20], CenterNet: Channel attention module [22] added with our proposed spatial attention module and backbone network, LogoNet, Dual-Attention LogoNet.

SSD achieves 76.6% accuracy in mAP with the faster detection time of 0.0531 s. Faster R-CNN has 81.0% accuracy with a 0.1115 s inference time. CenterNet with HourGlass achieves 80.7% accuracy and uses 0.1083 s detection time. There is a slight drop in the performance of CenterNet-HourGlass with SENet and ECANet block. These approaches have 80.2% and 79.0% accuracy with 0.1354 s and 0.1260 s detection time, respectively. Channel attention module [22] employed with our proposed spatial attention module and backbone network improves the accuracy by around 0.7% In comparison to baseline method. Whereas, detection time taken is relatively higher (0.2010 s per image) for this approach. LogoNet shows a significant improvement in performance over the conventional methods with a considerable detection time. LogoNet has 82.2% mAP accuracy with 0.1145 s inference time. Meanwhile, our proposed Dual-Attention LogoNet yields an improved performance with the 82.5% mAP and 0.1166 s detection time. The logo detection performance is depicted in Figure 8.



Figure 8. Visualization of multiple logos detection and effectiveness of our approaches.

Table 4. Performance Evaluation of State-of-the-Art Methods on the FlickrLogos-32 Dataset.

Methods	mAP	Detection Time
SSD [4]	76.7	0.0531 s
Faster-RCNN [2]	81.0	0.1115 s
CenterNet (baseline) [13]	80.7	0.1083 s
CenterNet (SENet [5])	80.2	0.1354 s
CenterNet (ECANet [15])	79.0	0.1260 s
CenterNet (CBAM [16])	81.4	0.2010 s
LogoNet	82.2	0.1145 s
Dual-Attention LogoNet	82.5	0.1166 s

Figure 9 shows the visualization of the last layer's feature maps of methods—CenterNet, CenterNet: ECANet, LogoNet, Dual-Attention LogoNet. These binary output images illustrate response of various attention-weight methods. Our spatial attention and dual-attention-based methods emphasize on logo objects and reduce the noise.

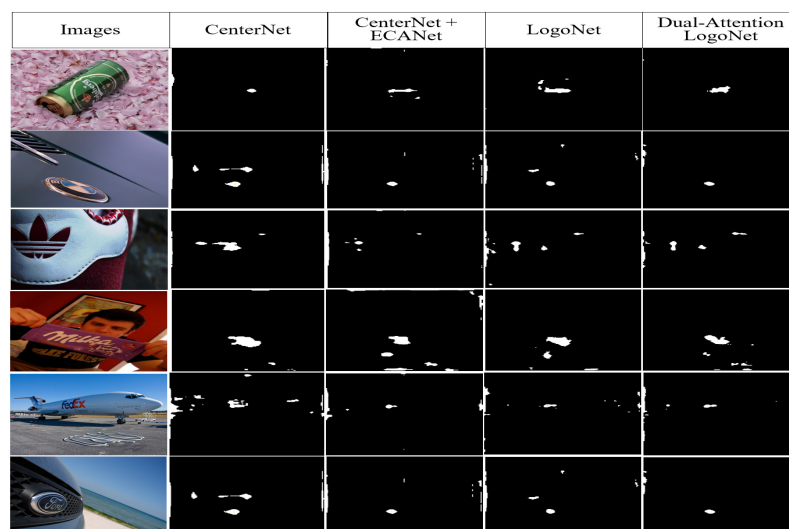


Figure 9. Visualizing different attention specific response for logo detection.

3.3. Evaluation on Logos-32plus

In [35] Logos-32plus is presented as an extended version of the FlickrLogos-32 dataset. It has 7830 training images for 32 logo classes (similar to FlickrLogos-32). To perform the experiments, we randomly split training images of each category into 90% as training and 10% as validation. Whereas, the official test set of FlickrLogos-32 is used. The author carefully created this dataset to include a comprehensive data distribution of real world logo images. Since the Logos-32plus dataset is 6 times larger than the FlickrLogos-32 dataset, performance on FlickrLogos-32 test set is notably increased. Results show that various characteristics such as dataset size, style and data distribution have a large impact on performance.

Table 5 gives the mAP and detection times for CenterNet and LogoNet. CenterNet achieves 88% mAP accuracy because the dataset has a significant data distribution for model learning. LogoNet delivers improved performance and has 88.3% mAP accuracy.

Table 5. Performance Evaluation of State-of-the-Art Detection Methods on Logos-32Plus Dataset.

Methods	mAP	Detection Time
CenterNet	88.0	0.1093 s
LogoNet	88.3	0.1156 s

3.4. Evaluation with Lightweight CNNs Method

We evaluated the proposed lightweight CNNs methods using the FlickrLogos-32 dataset. Detection accuracy in mAP, number of parameters in millions and image detection time in seconds are given in Table 6. For the Lightweight architectures, training has been conducted with a batch size of 4 for 140 epochs. The rest of the parameter setting is used as before. Due to the limited data, we initialize the network weights on PASCAL-VOC non-logo object detection images [36]. We observed that when we apply depthwise and pointwise convolution operation there is a drastic reduction in network parameters compared to the standard convolution operation. The reduction in parameters leads to faster computation speed but slightly declines detection accuracy. For comparison, we implemented CPD and Lightweight CNNs modules with CenterNet, LogoNet and Dual-Attention LogoNet.

CenterNet architecture based on CPD method (CenterNet-CPD) achieves 77.9% detection accuracy while the number of parameters is 72.42 million and detection time is 0.1145 s. LogoNet-CPD network achieves a greater accuracy 78.8% with detection time of 0.1073 s. However, Dual-Attention LogoNet network achieves 78.9% accuracy with a 0.1145 s detection time. LogoNet-CPD and Dual-Attention LogoNet-CPD use around 73.19 million computation parameters. With our lightweight modules, CenterNet (CenterNet-Lightweight) achieves 79.0% accuracy with 0.0833 s detection time. This architecture uses 27.94 million parameters. Dual-Attention LogoNet-Lightweight achieves 79.5% accuracy with a detection time of 0.0979 s. The proposed LogoNet-Lightweight network achieves a significantly higher accuracy rate of 79.7%, which is slightly less than the baseline (CenterNet) and LogoNet methods (80.7% and 82.2%). Whereas, LogoNet-Lightweight takes a detection time of 0.0885 s per image, which is about 20% faster than the baseline method. The LogoNet-Lightweight and Dual-Attention LogoNet architectures use only 28.73 million parameters. The parameters used are only about 15% of the parameters used in the normal baseline method (CenterNet). We found that LogoNet-Lightweight, which incorporates only spatial attention module, achieves a better performance in terms of detection time and accuracy. This approach leads to faster training and convergence of the network. Since depthwise convolution operations are used in the lightweight modules, channel-wise attention is not very effective. A model with low parameters and considerable accuracy rate is preferable for edge computing devices. We believe our lightweight algorithm is more suitable to run on low-spec machines or for edge computing than conventional algorithms.

Table 6. Performance Evaluation of the Lightweight Methods on FlickrLogos-32 Dataset.

Detectors	Lightweight Methods	mAP	Parameters	Detection Time
CenterNet [14]	No	80.7	191.26 M	0.1093 s
	CPD [33]	77.9	72.42 M	0.1145 s
	Proposed Method	79.0	27.94 M	0.0833 s
LogoNet [15]	No	82.2	192.05 M	0.1145 s
	CPD [33]	78.8	73.19 M	0.1073 s
	Proposed Method	79.7	28.73 M	0.0885 s
Dual-Attention LogoNet	No	82.5	192.05 M	0.1166 s
	CPD [33]	78.9	73.20 M	0.1145 s
	Proposed Method	79.5	28.73 M	0.0979 s

3.5. Evaluation with Adversarial-Based Domain Adaptation

To implement adversarial domain adaptation approach, we utilize the FlickrLogos-32 dataset [34] as source domain and Logos-32plus dataset [35] as target domain. The training images of target domain (i.e., Logos-32plus) are collected to represent a comprehensive real-world data presentation. These target domain images (Logos-32plus dataset) are captured on different platforms and in different sizes, shape, illumination and viewpoints, whereas most of the training images in the source domain (i.e., FlickrLogos-32) dataset are captured on plane and cylindrical surfaces and in selected viewpoints. The data distributions of these two datasets are very different from each other. The task of detection becomes very challenging when the model is trained on the source domain that has less comprehensive data representation and tested to the target domain that does not have the same distribution and style as source.

To perform the experiment under the domain-shift problem, we needed a test set with different domain representations. We randomly selected 30 images for each class from the target domain (Logos-32plus dataset) and created a new test set of 960 images, while the training set is source domain (FlickrLogos-32 dataset). The remaining images of target domain (Logos-32plus dataset) are used during the training. Note that in this experiment only the source domain (FlickrLogos-32 dataset) is annotated, while the target domain (Logos-32plus dataset) is not annotated. This is a case of domain shift (FlickrLogos-32 to Logos-32plus, scene adaptation) because both datasets have training images with different data distribution and styles. The details of the datasets are provided in Table 7. The target domain has a total of 7830 images, of which 6870 are considered as training (unlabeled) and 960 images are used as test set.

Table 7. Detail of training data setting.

Datasets	Labeled Training Images	Unlabeled Target Images	Test Images
FlickrLogos-32 (Source dataset)	960	-	-
Logos-32plus (Target dataset)	-	6870	960

The training parameters setting is used as before. The initial learning rate for the discriminant network is 0.0001 which is decreased by $\times 0.1$ at 90 and 120 epochs. The Adam optimizer is used. During training, the same batch-sized images from the source and target domains are used to train the model. In each epoch, only randomly selected 960 target images out of 6870 images have used for the training.

Table 8 shows the detection results of LogoNet: normal training, LogoNet: domain adaptation using class-wise heatmaps, and LogoNet: domain adaptation using Mid-level

feature maps (proposed approach). In our experiments, LogoNet trained in the normal setting achieves 63.2 mAP accuracy. In [37], the authors proposed to use class-wise heatmaps to adapt domain shift from synthetic to real images. In our case, we use class-wise heatmaps to implement adversarial domain adaptation. The heatmaps based domain adaptation achieves 59.6 mAP accuracy. According to the results, when heatmaps are used to align the domains, the accuracy is dramatically lower than the direct transfer of LogoNet method. This is for two reasons, first, class-wise heatmaps do not maintain important image level information. Second, anchorfree detectors train the network to detect objects in terms of keypoints, so this layer loses significant domain specific information. LogoNet with mid-level domain adaptation shows an improvement in performance by achieving 64.5 mAP accuracy. Our proposed method increases the performance by 1.3% mAP compared to the direct transfer of the detection network.

Table 8. Effectiveness of the domain adaptation on Logos-32plus dataset.

Methods	mAP
LogoNet [15] (w/o domain adaptation)	63.2
LogoNet + Domain Adaptation (Class-wise heatmaps)	59.6
LogoNet + Domain Adaptation (Proposed method - Mid-level feature maps)	64.5

Table 9 reports the comparison results for domain adaptation-based methods. To compare with other state-of-the-art methods, we train domain-adaptive Faster R-CNN [38] using our datasets. This approach uses a gradient reversal layer [39] to train the generator (backbone network) and the discriminator network. The backbone network is the fpn Resnet50 [30]. Scheck et al. [37] proposed to use entropy minimization loss [25] and maximum square loss [27] for the detection task. We used their network with the given parameter setting on our datasets. Domain adaptation using Faster R-CNN achieves 59.7 mAP accuracy for our dataset. Entropy minimization and maximum square loss-based networks achieve 59.4 mAP and 59.6 mAP accuracy, respectively. Our proposed method improves the detection performance and achieves a 64.5 mAP accuracy.

Table 9. Comparison with existing domain adaptation methods.

Methods	mAP
Scheck et al. [37] (Entropy Minimization Loss)	59.4
Scheck et al. [37] (Maximum Square Loss)	59.6
Hsu et. al. [38]	59.7
Proposed method	64.5

4. Discussion

In this paper, we performed logo detection using attention based mechanisms with an anchor-free detector for the logo datasets containing real-world images. The performance of our approach is evaluated with anchorfree detector CenterNet and anchorbox based detectors like SSD and Faster R-CNN. The experiments show that the CenterNet method is robust and faster with an 80.7% mAP. We propose a feature extractor network with spatial and channel attention modules to effectively capture information from complex logo images to fuse visual and semantic features. Our proposed approaches, LogoNet

and Dual-Attention LogoNet, provide significant detection capability with a considerable detection time and achieve better performance with 82.2% and 82.5% mAP, respectively. The proposed architecture can be learned to detect new sets of logo classes. Logo images include diverse context, illumination, resolutions that make logo detection a challenging task. A robust feature extractor that can emphasize and discriminate various logo regions would be more suitable. More attention-based methods can be used to generate refined feature maps. A logo detector can be trained by considering all logo classes as a single logo (in a class-agnostic way). In this case the logo detector will be able to detect and classify the logo regions as a general logo class [1]. We also proposed lightweight CNNs architecture to improve the real-time performance of network. We apply different lightweight modules with the proposed backbone and compared the networks. According to the experiments, LogoNet-Lightweight network achieves 79.7% accuracy, with a smaller number of parameters and reduced detection time. The proposed methods improve the focus on logos and detect logos more precisely than conventional algorithms. To bridge the gap between different domains we exploit the adversarial-domain adaptation learning. We propose a pragmatic way of dealing with the domain-shift problem using an anchorfree object detector. We make use of mid-level output feature maps to align the domains and to train a robust detector model. This approach can be easily be adapted to other anchorfree detectors. Training in adversarial manner is a difficult task for detection we will consider more approaches for better performance and stability.

5. Conclusions

We have proposed a Dual-Attention-based LogoNet Network, using spatial and channel attention modules. Our architecture refines output feature maps and improves the performance with an accuracy gain of 1.8% in a considerable computation time. Furthermore, we propose a lightweight CNNs method with anchor-free detector. We also propose an adversarial learning-based domain adaptation approach to align the detection network between source and target domains. In future, we will discover more attention- and domain adaptation-based mechanisms including transformer [40] and lightweight compact network for logo detection in real-time.

In this paper, we propose Dual-Attention LogoNet. The backbone architecture of the proposed method includes a densely layer-aggregated hourglass-like network. Spatial and channel attention modules are added to further refine the feature maps. The CenterNet detection head is used [14].

Our key contributions are as follows:

(1) We propose an attention-based architecture called LogoNet, which includes a backbone feature extraction framework that aggregates feature maps at different scales. This framework efficiently extracts feature information from different scales and also prevents loss of information during spatial resolution scaling.

(2) The proposed spatial attention module enhances attention to identify target objects. This attention module refines the output feature maps. It serves as a tool to focus on the logo regions.

A preliminary version of this work was presented as a five-page conference paper at IEEE International Conference on Consumer Electronics-2021 [15]. As an extension, here, we introduce a dual attention-based method by employing a channel attention module along with the spatial attention module, a lightweight CNN architecture, and a domain optimization-based approach. Our new contributions are as follows.

(3) The channel attention module is combined with the new proposed architecture in a different and effective manner to make it more efficient.

(4) We propose a lightweight CNNs architecture with a reduced number of network parameters and computation complexity. The architecture can boost the run-time associated with the inference of network while maintaining the performance.

(5) We propose an adversarial learning-based domain adaptation approach to generalize the network from source to target domain. We propose to use the mid-level output

feature maps of the feature extraction network instead of using class-wise heatmaps, which is commonly used in most of the previously proposed domain adaptation based methods. To the best of our knowledge, this is the first domain discriminator network-based adversarial learning scheme employed with an anchor-free detector.

Author Contributions: Conceptualization, R.K.J., Y.-W.C., X.R.; methodology, software, validation, formal analysis, investigation, R.K.J., T.W., T.N., T.S.; resources, Y.-W.C.; data curation, R.K.J.; writing—original draft preparation, R.K.J.; writing—review and editing, visualization, Y.-W.C., X.R., Y.I.; supervision, Y.-W.C., X.R.; project administration, Y.-W.C., X.R.; funding acquisition, Y.-W.C., X.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fehérvári, I.; Appalaraju, S. Scalable logo recognition using proxies. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 715–725.
2. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duna, Y.; Al-Shamma, O.; Santamaria, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *J. Big Data* **2021**, *53*, 1–74.
3. Xiao, Y.; Tian, Z.; Yu, J.; Zhang, Y.; Liu, S.; Du, S.; Lan, X. A review of object detection based on deep learning. *Multimed. Tools Appl.* **2020**, *79*, 23729–23791. [[CrossRef](#)]
4. Chen, Y.W.; Jain, L.C. (Eds.) *Deep Learning in Healthcare*; Springer: Berlin/Heidelberg, Germany, 2020.
5. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)] [[PubMed](#)]
6. Redmon, J.; Farhadi, A. Yolo9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the ECCV, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
8. Hu, J.; Shen, L.; Sun, G. Squeeze and excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7132–7142.
9. Su, H.; Zhu, X.; Gong, S. Deep learning logo detection with data expansion by synthesising context. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Santa Rosa, CA, USA, 24–31 March 2017; pp. 530–539.
10. Su, H.; Zhu, X.; Gong, S. Open logo detection challenge. In Proceedings of the British Machine Vision Conference, Newcastle, UK, 3–6 September 2018.
11. Su, H.; Zhu, X.; Gong, S. Weblogo-2m: Scalable logo detection by deep learning from the web. In Proceedings of the International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2017; pp. 270–279.
12. Su, H.; Zhu, X.; Gong, S. Scalable logo detection by self co-learning. *Pattern Recognit.* **2020**, *97*, 107003. [[CrossRef](#)]
13. Jain, R.K.; Iwamoto, Y.; Watasue, T.; Nakagawa, T.; Sato, T.; Ruan, X.; Chen, Y.W. Weakly Supervised Logo Detection Using a Dual-Attention Dilated Residual Network. *IEEE Trans. Image Electron. Vis. Comput.* **2021**, *9*, 15–22.
14. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. *arXiv* **2019**, arXiv:1904.07850.
15. Jain, R.K.; Watasue, T.; Nakagawa, T.; Sato, T.; Iwamoto, Y.; Ruan, X.; Chen, Y.W. LogoNet: Layer-Aggregated Attention CenterNet for Logo Detection. In Proceedings of the IEEE International Conference on Consumer Electronics, Las Vegas, NV, USA, 10–12 January 2021.
16. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *Int. J. Comput. Vis.* **2019**, *128*, 642–656. [[CrossRef](#)]
17. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. CenterNet: Keypoint Triplets for Object Detection. In Proceedings of the International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 6568–6577.
18. Zhou, X.; Zhou, J.; Krähenbühl, P. Bottom-up Object Detection by Grouping Extreme and Center Points. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 850–859.
19. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6450–6458.
20. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020.
21. Chen, X.; Lin, L.; Liang, D.; Hu, H.; Zhang, Q.; Iwamoto, Y.; Han, X.H.; Chen, Y.W.; Tong, R.; Wu, J. A dual-attention dilated residual network for liver lesion classification and localization on CT images. In Proceedings of the IEEE International Conference on Image Processing, Taipei, Taiwan, 22–25 September 2019; pp. 235–239.
22. Woo, S.; Park, J.; Lee, J.Y. CBAM: Convolutional Block Attention Module. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018; Springer: Cham, Switzerland, 2018; pp. 3–19.

23. Zhu, F.; Li, H.; Quyang, W.; Yu, N.; Wang, X. Learning Spatial Regularization With Image-Level Supervisions for Multi-Label Image Classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 2027–2036.
24. Saito, K.; Kohei, W.; Ushiku, Y.; Harada, T. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3723–3732.
25. Vu, T.H.; Jain, H.; Bucher, M.; Cord, M.; Perez, P. ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2512–2521.
26. Shen, R.; Yao, J.; Yan, K.; Tian, K.; Jiang, C.; Zhou, K. Unsupervised domain adaptation with adversarial learning for mass detection in mammogram. *Neurocomputing* **2020**, *393*, 27–37. [\[CrossRef\]](#)
27. Chen, M.; Hongyang, X.; Cai, D. Domain Adaptation for Semantic Segmentation with Maximum Squares Loss. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 2090–2099.
28. Newell, A.; Yang, K.; Deng, J. Stacked Hourglass Networks for Human Pose Estimation. In Proceedings of the ECCV, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 483–499.
29. Yu, F.; Wang, D.; Shelhamer, E.; Darrell, T. Deep Layer Aggregation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2403–2412.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
31. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2999–3007.
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
33. Lebedev, V.; Ganin, Y.; Rakhuba, M.; Oseledets, I.; Lempitsky, V. Speeding-up convolutional neural networks using fine-tuned cpdecomposition. *arXiv* **2014**, arXiv:1412.6553.
34. Romberg, S.; Pueyo, L.G.; Leinhardt, R.; Zwol, R.V. Scalable logo recognition in real-world images. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, New York, NY, USA, 18–22 April 2011; p. 25.
35. Binaco, S.; Buzzelli, M.; Mazzini, D.; Schettni, R. Deep Learning for Logo Recognition. *Neurocomputing* **2017**, *245*, 23–30. [\[CrossRef\]](#)
36. Everingham, M.; Gool, L.V.; Williams, C.K.I.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [\[CrossRef\]](#)
37. Scheck, T.; Grassi, A.P.; Gangolf, H. Unsupervised Domain Adaptation from Synthetic to Real Images for Anchorless Object Detection. In Proceedings of the Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), online, 8–10 February 2021.
38. Hsu, H.K.; Yao, C.H.; Tesg, H.Y.; Yao, C.H.; Tsai, Y.H.; Singh, M.; Yang, M.H. Progressive Domain Adaptation for Object Detection. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020.
39. Gannin, Y.; Lempitsky, V. Unsupervised domain adaptation by backpropagation. In Proceedings of the the 32nd International Conference on International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 1180–1189.
40. Dosovitskiy, A.; Beyar, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16X16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2021.