WW-Nets: Dual Neural Networks for Object Detection

Mohammad K. Ebrahimpour*, J. Ben Falandays+, Samuel Spevack+, Ming-Hsuan Yang*, and David C. Noelle*+
*Electrical Engineering & Computer Science, +Cognitive and Information Sciences
University of California, Merced

***Abstract*—We propose a new deep convolutional neural network framework that uses object location knowledge implicit in network connection weights to guide *selective attention* in object detection tasks. Our approach is called What-Where Nets (WW-Nets), and it is inspired by the structure of human visual pathways. In the brain, vision incorporates two separate streams, one in the temporal lobe and the other in the parietal lobe, called the ventral stream and the dorsal stream, respectively. The ventral pathway from primary visual cortex is dominated by "what" information, while the dorsal pathway is dominated by "where" information. Inspired by this structure, we have proposed an object detection framework involving the integration of a "What Network" and a "Where Network". The aim of the What Network is to provide selective attention to the relevant parts of the input image. The Where Network uses this information to locate and classify objects of interest. In this paper, we compare this approach to state-of-the-art algorithms on the PASCAL VOC 2007 and 2012 and COCO object detection challenge datasets. Also, we compare out approach to human "ground-truth" attention. We report the results of an eye-tracking experiment on human subjects using images from PASCAL VOC 2007, and we demonstrate interesting relationships between human overt attention and information processing in our WW-Nets. Finally, we provide evidence that our proposed method performs favorably in comparison to other object detection approaches, often by a large margin.**

***Index Terms*—Object Detection, Selective Attention, Deep Neural Networks**

I. Introduction

In recent years, deep Convolutional Neural Networks (CNNs) have been shown to be effective at image classification, accurately performing object recognition even in cases involving a large array of object classes, given a sufficiently rich dataset of images [1]–[4].

Image classification is only one of the core problems of computer vision, however. Beyond object recognition [2]–[4], there are applications for such capabilities as semantic segmentation [5]–[7], image captioning [8]–[10], and object detection [11]–[14]. The last of these involves locating and classifying all of the relevant objects in an image. This is a challenging problem that has received a good deal of attention [11]–[13], [15]. Since there is rarely *a priori* information about where objects are located in an image, most approaches to object detection conduct exhaustive searches over image regions, seeking objects of interest with different sizes and aspect ratios. For example, region proposal frameworks, like Faster-RCNN [12], need to pass a large number of candidate image regions through a deep network in order to determine which parts of the image contain the most information concerning objects of interest. An alternative approach involves one shot detectors, like Single Shot Detectors (SSD) [16] and You Only Look Once (YOLO) [13]. These methods use networks to examine all parts of the image via a tiling mechanism. For example, YOLO conducts a search over potential combinations of tiles. The hope of single shot detection approaches is to find the responsible tile for the object and then identify the appropriate object location bounding box around that tile. In a sense, most off-the-shelf object detection algorithms distribute attention to all parts of the image equally. Allocating equal attention across the scene is not common in humans, however. Our brain has evolved pay selective attention to the vital parts of the visual scene [17].

Recently, an object detection framework inspired by the human visual system has been proposed, called Ventral-Dorsal Nets or, simply, VDNets [18]. In VDNets, a sensitivity analysis in a pretrained image classification network (the Ventral Net) is used to guide localization, and this approach shows promise. However, mistakes made by the Ventral Net can be catastrophic, masking out image regions that contain objects of interest, making their detection impossible. Moreover, despite the fact that this approach is accurate and faster than typical region proposal based object detection algorithms, it still cannot process images in real time.

In this paper, we propose a new method for extracting dense information about object location from pretrained networks, with the goal of improving selective attention. Specifically, these new What-Where Nets (WW-Nets) make use of channel-wise attention levels, as well as spatially specific attentional information from all receptive fields. By stacking these different kinds of attention maps, we can potentially preserve the semantic (object class) information that comes from late layers in the network while incorporating spatial location information that is richly represented in early layers. By using these stacked attention maps [4], our selective attention method is substantially different than a simple sensitivity analysis. We have found that our selective attention mechanism can also substantially improve object detection performance. Moreover, we trained a single shot detector on top of our salience map mechanism, which made the WW-Nets suitable for real time applications.

The human visual attention system supports our naturally

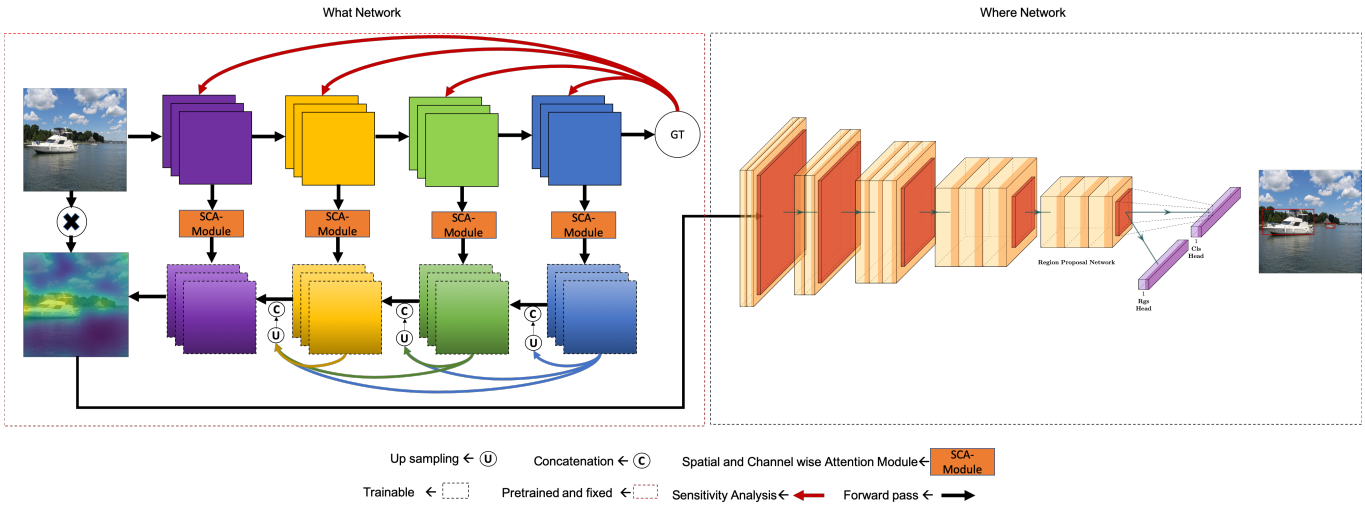

Fig. 1. WW-Nets architecture. The What Net guides the selective attention via combining the Spatial and Channel wise Attention in all convolutional layers (SCA). It also leverages from the hidden semantic information in the late layers as well as the hidden location information in the earlier layers by dense connections. Then the filtered image feeds into the Where Net for drawing the bounding boxes out of all objects of interest.

strong visual perception capabilities, so we see it as a useful guide for assessing selective attention behavior. Thus, in addition to measuring the object detection performance of our proposed system, we investigated possible relationships between information in WW-Nets and the patterns of attention exhibited by humans observing the same images. We used the distribution of fixation points produced by human subjects, measured using eye-tracking technology, as a measure of how the visual system distributes attention over images. The human visual attention data will be available publicly for future research efforts. The contribution of this work can be summarized as follows:

- Our WW-Nets include a sophisticated, learning free, mechanism to obtain information from different levels of the CNN. It assigns a weight to every single spatial location and channel in every convolutional layer. Also, it leverages the information from the most abstract features as well as the hidden location information in the earlier layers by stacking them together.
- We report human attention data collected via an eye-tracking study to provide "ground-truth" information for vision researchers interested in where people look.
- We show that, when identifying objects in benchmark image datasets, this framework provides superior object detection performance over comparison methods, often by a large margin.
- We provide comparisons of the attention of WW-Nets to human attention.

II. Related Work

Attention-Based Object Detection. Attention-based object detection methods depend on a set of training images with associated class labels but *without* any object location bounding box annotations. The lack of a need for ground-truth bounding boxes is a substantial benefit of this approach, since manually obtaining such information is costly.

One object detection approach of this kind is the Class Activation Map (CAM) method [19]. This approach is grounded in the observation that the fully connected layers that appear near the output of typical CNNs largely discard spatial information. To compensate for this, the last convolutional layer is scaled up to the size of the original image, and Global Average Pooling (GAP) is applied to the result. A linear transformation from the Global Average Pooling (GAP) values to class labels is learned. The learned weights to a given class output are taken as indicating the relative importance of different filters for identifying objects of that class.

For a given image, the individual filter activation patterns in the upscaled convolutional layer are entered into a weighted sum, using the linear transformation weights for a class of interest. The result of this sum is a Class Activation Map that reveals image regions associated with the target class.

The object detection success of the CAM method has been demonstrated, but it has also inspired alternative approaches. The work of Selvaraju et al. [20] suggested that Class Activation Maps could be extracted from standard image classification networks without any modifications to the network architecture and additional training to learn filter weights. The proposed Grad-CAM method computes the gradients of output labels with respect to the last convolutional layer, and these gradients are aggregated to produce the filter weights needed for CAM generation. This is an excellent example of saliency based approaches that interpret trained deep CNNs, with others also reported in the literature [21]–[23].

As previously noted, attention-based object detection methods benefit from their lack of dependence on bounding box annotations on training images. They also tend to be faster than supervised object detection approaches, producing results by interpreting the internal weights and activation

maps of an image classification CNN. However, these methods have been found to be less accurate than supervised object detection techniques.

Supervised Object Detection. Supervised object detection approaches require training data that include both class labels and tight bounding box annotations for each object of interest. Explicitly training on ground-truth bounding boxes tends to make these approaches more accurate than weakly supervised methods. These approaches tend to be computationally expensive, however, due to a need to search through the space of image regions, processing each region with a deep CNN. Tractability is sought by reducing the number of image regions considered, selecting from the space of all possible regions in an informed manner. Methods vary in how the search over regions is constrained.

Some algorithms use a region proposal based framework. A deep CNN is trained to produce both a classification output and location bounding box coordinates, given an input image. Object detection is performed by considering a variety of rectangular regions in the image, training the CNN class output when an object of interest is in the input region presented to the network. Importantly, rather than consider all possible regions, the technique depends on a *region proposal algorithm* to identify the image regions to be processed by the CNN. The region proposal method could be either an external algorithm like Selective Search [24], or it could be an internal component of the network, as done in Faster-RCNN [12]. The most efficient object detection methods of this kind are R-CNN [15], Fast-RCNN [11], Faster-RCNN [12], and Mask-RCNN [7]. Approaches in this framework tend to be quite accurate, but they face a number of challenges beyond issues of speed. For example, in an effort to propose regions containing objects of known classes, it is common to base region proposals on information appearing late in the network, such as the last convolutional layer. The lack of high resolution spatial information late in the network makes it difficult to detect small objects using this approach. There are a number of research projects that aim to address this issue by combining low level features and high level ones in various ways [21], [25].

Rather than incorporating a region proposal mechanism, some supervised methods perform object detection in one feed-forward pass. A prominent method of this kind is YOLO, as well as its extensions [13], [14]. In this approach, the image is divided into tiles, and each tile is annotated with anchor boxes of various sizes, proposing relevant regions. The resulting information, along with the image tiles, are processed by a deep network in a single pass in order to find all objects of interest. While this technique is less accurate than region proposal approaches like Faster-RCNN, it is much faster, increasing its utility for online applications.

It is worth noting that supervised object detection methods can be seen as spreading attention across the full image, examining all possible regions, to some degree. This is computationally costly.

A comparison of these two general approaches to object detection displays a clear trade-off between accuracy and computational cost (speed). This gives rise to the question of whether this trade-off can be avoided, in some way.

Dual Neural Networks as Two Pathways for Object Detection. As previously noted, our brain evolved to pay attention to the gist of the scene and ignore the unimportant parts [17]. This is one of the reasons why human beings are good at finding objects in images. Neither of the two general frameworks for object detection take advantage of such a mechanism. Recently, a novel object detection framework called Ventral-Dorsal Networks (VDNets) has been proposed as an object detection approach inspired by the human visual system. VDNets are actually composed of two interacting deep networks, reflecting the two major information processing streams emerging from primary visual cortex [18]. We have found that VDNets exhibit strong object detection performance, but they can fail catastrophically if the Ventral Network mistakenly masks out objects of interest. Moreover, despite an increase in detection speed, this method cannot be used in real time applications due to a bottleneck in the Dorsal Network.

With the goal of improving Ventral Network performance, this paper proposes a substantially different selective attention algorithm. Rather than relying on an assessment of the contribution of pixels to the output of the last convolutional layer of a pretrained image classification network, we propose performing sensitivity analyses at all of the layers in the network and densely aggregating the resulting information to guide selective attention. This approach is intended to make use of both semantic information and spatial information by considering channel-wise attention and spatial attention at every convolutional layer in a pretrained classification CNN called the What Network. The channel-wise and spatial sensitivity values for each layer are stacked to preserve object location information implicit in the full network. The result is used to guide selective attention, masking the input image before it is presented to the Where Network for object detection. We also examined architectures for the Where Network appropriate for real time tasks. Finally, we compared this attention mechanism to human data.

III. Algorithm

Dense Attention of What Networks. Our investigations into WW-Nets used a pretrained image classification network to guide selective attention. In our initial studies, for a given input image, the sensitivity of activity late in the network to each pixel was efficiently calculated, and regions containing low sensitivity pixels were masked out. Our What Network has two major parts. First, we calculated the saliency of image regions based on activity at layers throughout the network, rather than only at the last convolutional layer. We aggregated spatial and channel-wise sensitivity information at each layer to produce layer-specific attention maps. Second, we densely stacked the whole collection of attention maps, from all layers to capture the most abstract features of the

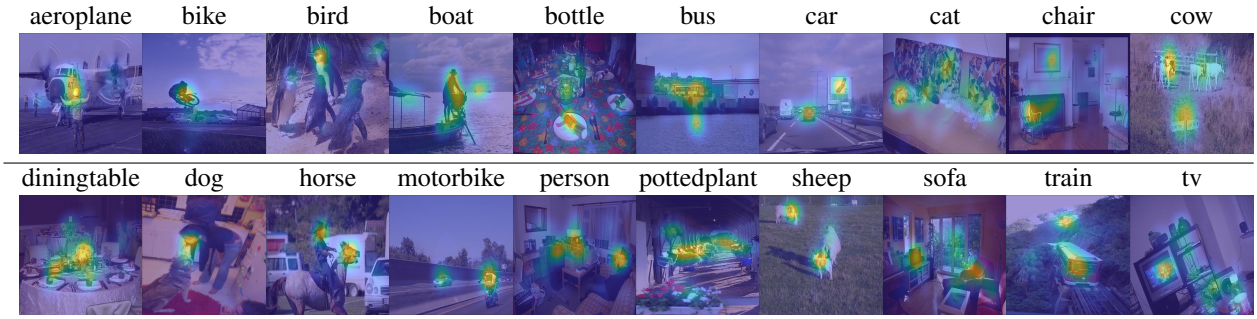

Fig. 2. Example Human Eye Fixation Distributions on PASCAL VOC Images

objects in the deep layers as well as obtaining the spatial locations of the objects in the shallow layers, in order to drive selective attention to image regions. The architecture of WW-Nets is depicted in Figure 1.

A. Spatial & Channel-wise Attention (SCA) Module

Spatial Attention. Generally speaking, objects occupy only parts of images, leaving background regions that can distract and misinform object detection systems. Instead of considering all parts of an image equally, spatial attention can focus processing on foreground regions, supporting the extraction of features most relevant for object class and object extent.

We represent a convolutional feature of layer n by $f_n \in \mathbb{R}^{W\times H\times C}$, where W and H are the spatial dimensions of the rectangular layer and C is the number of feature channels in the layer. Spatial positions are specified by coordinate pairs: $\mathbb{L} = \{(x, y)|x = 1, 2, \ldots, W; y = 1, 2, \ldots, H\}$.

For a layer in a pretrained image classification network, the layer-specific spatial attention map is generated by the following equation:

$$A_n^s = W_n^s \odot f_n \tag{1}$$

where W_n are weights that indicate the importance of each spatial location, across all of the convolutional filters, for the current image. We initially calculate these weights based on the sensitivity of the Gestalt Total (GT) activation of the network to the feature. The Gestalt Total is calculated from the activation of the last convolutional layer, A_{last}, as follows:

$$GT = \frac{1}{H \times W \times C} \sum_{i,j,k} A_{last}(i, j, k) \tag{2}$$

GT is the activation over the most abstract features in the last convolutional layer. The sensitivity of GT to a feature at layer n is the following:

$$G_n = \frac{\partial GT}{\partial f_n};\ W_n^u(x, y) = \sum_c^C G_n(x, y, c) \tag{3}$$

where W_n^u is not normalized (i.e., the weights are not in the $[0, 1]$ range). To normalize the weights for each location, ℓ, we apply a softmax operation to the weights spatially:

$$W_n^s(l) = \frac{\exp(W_n^u(l))}{\sum_{i\in\mathbb{L}} \exp(W_n^u(i))} \tag{4}$$

where $W_n^s(l)$ denotes the weight for location l in layer n.

Channel-Wise Attention. The spatial attention calculation assigns weights to spatial locations, which addresses the problem of distractions from background regions. There is another way in which distractions can arise, however. Specific channels at a given layer can be distracting. When dealing with convolutional features, most existing methods treat all channels without distinction. However, different channels are often different in their relevance for objects of specific classes. Here, we introduce a channel-wise attention mechanism that assigns larger weights to channels to which the GT is sensitive, given the currently presented image. Incorporating these channel-wise attentional weights are intended to reduce this kind of distracting interference.

For channel-wise attention, we unfold f_n as $f = [f_n^1, f_n^2, \ldots, f_n^C]$, where $f_n^i \in \mathbb{R}^{W\times H}$ is the i$^{\text{th}}$ slice of f_n, and C is the total number of channels. The goal is to calculate a weight, W_n^c, to scale the convolutional features according to a channel-specific assessment of relevance:

$$A_n^c = W_n^c \cdot f_n \tag{5}$$

Computing W_n^c is facilitated by the fact that we already have the sensitivities, G_n. Thus, an initial value for the weights can be had by setting $\hat{W}_n^c(c) = \sum_{x\in W, y\in H} G_n(x, y, c)$. These weights can be normalized to the $[0, 1]$ range using the softmax function:

$$W_n^c(c) = \frac{\exp(\hat{W}_n^c(c))}{\sum_{i\in C} \exp(\hat{W}_n^u(i))} \tag{6}$$

These are the final channel-wise weights for layer n.

Dense Attention Maps. Given the spatial attention weights and the channel-wise attention weights, an attention weighted feature for layer n is calculated as:

$$f_n^{SCA} = A_n^c \cdot f_n + A_n^s \odot f_n \tag{7}$$

where $f_n^{SCA} \in \mathbb{R}^{W\times H\times C}$. These weighted features are concatenated across layers, incrementally from the last layer to the first, but this is done after up-scaling lower spatial resolution layers. For the last layer, m, the map is simply the weighted features: $A_m = f_m^{SCA}$. For earlier layers:

$$\begin{aligned} A_i = [UP(f_m^{SCA}), UP(f_{m-1}^{SCA}), \ldots, f_i^{SCA}] \\ i = \{1, \ldots, m-1\} \end{aligned} \tag{8}$$

The result is dense combination maps that are intended to incorporate both semantic information from the late layers and spatial information from the early layers. These attention maps are then combined to make a single map for the whole image. Since each layer can have a different number of channels, we simplify this aggregation by averaging each layer's attention map across channels, tranforming each A_i into a $W \times H$ matrix. The aggregated attention maps at each layer are computed as:

$$Att_i = A_m^{SCA} \oplus A_{m-1}^{SCA} \oplus \ldots \oplus A_i^{SCA} \tag{9}$$

The process of spatial & channel wise attention, along with dense connections among them, is depicted in Figure 1.

The final aggregated attention map, at the first layer, is Att_1. We smooth this $W \times H$ map by convolving it with a Gaussian filter, and then we threshold the result. This produces a binary mask specifying regions of relevance. The original image is multiplied by this mask to produce the input to the Where Network.

It is important to note that the attention weights (W_n^s and W_n^c) are not learned as part of a training process. We begin with a pretained image classification network for the Ventral Network, and the attention weights are efficiently calculated for each layer when a given image is presented to that network.

Where Network. Blanking irrelevant regions in the image substantially reduces the space of candidate regions to consider during object detection. In the Where Network, the masked image is provided as input to a deep CNN trained to propose regions of interest with anchor boxes, process the contents of those regions, and output both class labels and bounding box coordinates. This is similar to the approach used by Faster-RCNN [12]. The Where Net is trained using a dataset of images that are annotated with both ground-truth class labels and ground-truth bounding boxes. Network parameters are selected so as to minimize a combination of the classification loss and the regression loss arising from the output of bounding box coordinates:

$$\begin{aligned} L(p_i, t_i) = & \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) \\ & + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \end{aligned} \tag{10}$$

where i is the index of an anchor box appearing in the current training mini-batch and p_i is the predicted probability of anchor i containing an object of interest. The ground-truth label p_i^* is 1 if anchor i is positive for object presence and it is 0 otherwise. The predicted bounding box is captured by the 4 element vector t_i, and t_i^* contains the coordinates of the ground-truth bounding box associated with a positive anchor. The two components of the loss function are normalized by N_{cls} and N_{reg}, and they are weighted by a balancing parameter, λ. In our current implementation, the classification loss term is normalized by the mini-batch size (i.e., $N_{cls} = 32$) and the bounding box regression loss term is normalized by the number of anchor locations (i.e., $N_{reg} \approx 2,400$). We set $\lambda = 10$, making the two loss terms roughly equally weighted (due to differences in scale).

It is worth noting that our general approach could easily incorporate other object detection algorithms for the Where Net. The only requirement is that the object detection algorithm must be able to accept masked input images. For the results presented in this paper, we have used a region proposal based approach, due to the high accuracy values reported for these methods in the literature. Having the What Net reduce the number of proposed regions was expected to speed the object detection process and also potentially improve accuracy by removing from consideration irrelevant portions of the image. As discussed in the next section, we also trained YOLO as an alternative Where Net for achieving real time performance.

IV. Experimental Results

Experiment Design and Implementation. We evaluated the WW-Nets object detection model on PASCAL VOC 2007 [26], PASCAL VOC 2012 [27], and COCO datasets [28]. We also compared the attention model to human performance. Source code and human selective attention heatmaps will be made publicly available.

The PASCAL VOC 2007 dataset has 20 classes and $9,963$ images which have been equally split into a training/validation set and a test set. The PASCAL VOC 2012 dataset contains $54,900$ images from 20 different categories, and it has been split approximately equally into a training/validation set and a test set. For PASCAL VOC 2007, we conducted training on the union of the VOC 2007 trainval set and the VOC 2012 trainval set, and we evaluated the results using the VOC 2007 test set. (This regimen is standard practice for these datasets.) For PASCAL VOC 2012, we performed training on its trainval set, and we evaluated the result on its test set. To evaluate performance, we used the standard mean average precision (mAP) measure. We report mAP scores using IoU thresholds at 0.5 for PASCAL datasets and we used mAP with both IOU thresholds at 0.5 and 0.75 for the COCO dataset.

For networks with 224×224 image inputs, using PASCAL VOC, we trained the model with a mini-batch size of 1 due to GPU memory constraints. We started the learning rate at 3×10^{-4} for the first $900,000$ iterations. We then decreased it to 3×10^{-5} until iteration $1,200,000$. Then, we decreased it to 3×10^{-6} until iteration $2,000,000$. In all cases, we used a momentum optimizer value of 0.9.

PASCAL VOC 2007 Results. The results of the PASCAL VOC 2007 dataset evaluation appear in Table I. For the What Net, we utilized VGG16 pretrained on the ImageNet dataset [37]. We removed the fully connected layers and softmax calculation from VGG16, and we calculated GT based on the last convolutional layer. No fine tuning of parameters was done. For the Where Net we used Resnet 101 [30].

We also tried YOLO V2. All of the networks were trained using 4 GPUs.

We compared our performance results with those reported for a variety of state-of-the-art approaches to object detection. Our primary baseline was Faster-RCNN using a Resnet 101

TABLE I

PASCAL VOC 2007 TEST DETECTION RESULTS. NOTE THAT THE MINIMUM DIMENSION OF THE INPUT IMAGE FOR FASTER AND R-FCN IS 600, AND THE SPEED IS LESS THAN 10 FRAMES PER SECOND. SSD300 INDICATES THE INPUT IMAGE DIMENSION OF SSD IS 300×300. LARGE INPUT SIZES CAN LEAD TO BETTER RESULTS, BUT THIS INCREASES RUNNING TIMES. ALL MODELS ON THE UNION OF THE TRAINVAL SET FROM VOC 2007 AND VOC 2012 AND TESTED ON THE VOC 2007 TEST SET, EXCEPT FOR THE MODEL LABELED WW-NETS* WHICH HAS BEEN TRAINED ON THE TRAINVAL SET FROM VOC 2007 AND TESTED ON 2007 TEST SET. ALSO, WW-NETS$^+$ IS TRAINED ON THE UNION OF TRAINVAL SET FROM VOC 2007 AND VOC 2012 AND IMAGENET AND TESTED ON THE VOC 2007 TEST SET.

Method	Network	mAP	areo	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Faster [12]	VGG	73.2	76.5	79	70.9	65.5	52.1	83.1	84.7	86.4	52	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83	72.6
ION [29]	VGG	75.6	79.2	83.1	77.6	65.6	54.9	85.4	85.1	87	54.4	80.6	73.8	85.3	82.2	82.2	74.4	47.1	75.8	72.7	84.2	80.4
Faster [30]	Residual-101	76.4	79.8	80.7	76.2	68.3	55.9	85.1	85.3	89.8	56.7	87.8	69.4	88.3	88.9	80.9	78.4	41.7	78.6	79.8	85.3	72
MR-CNN [31]	VGG	78.2	80.3	84.1	78.5	70.8	68.5	88	85.9	87.8	60.3	85.2	73.7	87.2	86.5	85	76.4	48.5	76.3	75.5	85	81
R-FCN [32]	Residual-101	80.5	79.9	87.2	81.5	72	69.8	86.8	88.5	89.8	67	88.1	74.5	89.8	90.6	79.9	81.2	53.7	81.8	81.5	85.9	79.9
SSD300 [16]	VGG	77.5	79.5	83.9	76	69.6	50.5	87	85.7	88.1	60.3	81.5	77	86.1	87.5	83.9	79.4	52.3	77.9	79.5	87.6	76.8
SSD512 [16]	VGG	79.5	84.8	85.1	81.5	73	57.8	87.8	88.3	87.4	63.5	85.4	73.2	86.2	86.7	83.9	82.5	55.6	81.7	79	86.6	80
DSSD321 [33]	Residual-101	78.6	81.9	84.9	80.5	68.4	53.9	85.6	86.2	88.9	61.1	83.5	78.7	86.7	88.7	86.7	79.7	51.7	78	80.9	87.2	79.4
DSSD513 [33]	Residual-101	81.5	86.6	86.2	82.6	74.9	62.5	89	88.7	88.8	65.2	87	78.7	88.2	89	87.5	83.7	51.1	86.3	81.6	85.7	83.7
STDN300 [21]	DenseNet-169	78.1	81.1	86.9	76.4	69.2	52.4	87.7	84.2	88.3	60.2	81.3	77.6	86.6	88.9	87.8	76.8	51.8	78.4	81.3	87.5	77.8
STDN321 [21]	DenseNet-169	79.3	81.2	88.3	78.1	72.2	54.3	87.6	86.5	88.8	63.5	83.2	79.4	86.1	89.3	88.0	77.3	52.5	80.3	80.8	86.3	82.1
STDN513 [21]	DenseNet-169	80.9	86.1	89.3	79.5	74.3	61.9	88.5	88.3	89.4	67.4	86.5	79.5	86.4	89.2	88.5	79.3	53.0	77.9	81.4	86.6	85.5
VDNet [18]	Resnet-101	86.2	95.8	98.1	98.4	65.1	94.6	90.1	96.2	71.7	72.3	54.6	97.9	95.6	89.2	90.1	93.2	69.1	89.2	82.1	93.4.6	74.0
WW-Nets*	YOLONet	61.4	66.5	69.5	61.9	39.2	42.8	69.2	65.3	79.3	44.2	57.3	54.8	72.9	77.1	69.1	72.4	34.2	49.5	63.1	76.1	65.2
WW-Nets	YOLONet	63.7	80.1	73.2	54.1	47.2	43.1	74.5	73.2	78.6	42.1	62.8	57.3	74.9	77.7	73.6	73.2	30.2	53.1	64.1	75.9	65.9
WW-Nets	Resnet-101	**86.7**	95.8	98.4	98.2	66.4	94.6	90.2	96.1	71.2	72.8	54.9	97.9	95.6	89.6	90.3	93.2	69.6	89.2	82.1	93.2	74.1
WW-Nets$^+$	Resnet-101	**88.1**	94.1	97.7	98.9	67.7	94.5	92.1	95.9	72.1	73.2	52.2	98.0	96.3	87.9	91.4	91.8	70.6	91.1	81.9	95.1	72.6

TABLE II

PASCAL VOC 2012 TEST DETECTION RESULTS. WW-NETS$^+$ IS TRAINED ON THE UNION OF TRAINVAL VOC 2012 AND IMAGENET.NOTE THAT THE PERFORMANCE OF WW-NETS IS ABOUT 6% BETTER THAN BASELINE FASTER-RCNN.

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
HyperNet-VGG [34]	71.4	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.97	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7
HyperNet-SP [34]	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO [13]	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	89.4	49.4	75.5	57.0	87.5	80.9	81.0	74.7	41.8	71.5	68.5	82.1	67.2
MR-CNN-S-CNN [35]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [12]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
NoC [36]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
VDNets [18]	73.2	85.1	82.4	73.6	57.7	61.2	79.2	77.1	85.5	54.9	79.8	61.4	87.1	83.6	81.7	77.9	45.6	74.1	64.9	80.3	73.1
WW-Nets	**74.1**	85.7	82.2	74.1	55.9	60.9	79.8	76.4	83.5	56.9	78.4	63.4	88.4	83.9	81.3	77.1	46.5	74.6	65.3	80.1	72.9
WW-Nets$^+$	**76.8**	87.2	81.9	75.7	56.3	61.9	76.2	76.1	85.1	55.7	79.8	62.5	87.2	82.8	82.4	79.5	43.2	73.1	64.1	81.7	73.2

network trained on PASCAL VOC 2007. As shown in Table I, the selective attention process of our approach (What-Whare Nets) resulted in substantially better performance in comparison to Faster-RCNN and other methods. WW-Nets appear to be more accurate at detecting larger objects than smaller ones, perhaps because the region proposal network based its output on the last (lowest resolution) convolutional layer. For most of the object classes, WW-Nets performed favorably against other methods by a large margin. One shot detectors did not perform as well as the region proposal based approaches (as expected). We observed this by training YOLO V2 on PASCAL VOC 2007 and the union of PASCAL VOC 2007 and 2012 dataset and using the trained YOLO V2 network for the Where Net. The difference likely reflects a trade off between speed and accuracy.

PASCAL VOC 2012 Results. We also measured WW-Nets performance on the PASCAL VOC 2012 dataset. The What Net was VGG16, with features for calculating GT extracted from the last convolutional layer. The Where Net was initialized with parameters previously learned for the PASCAL VOC 2007 evaluation, but further training was done on PASCAL VOC trainval sets. For the additional training, the learning rate was initialized to 3×10^{-4} for $900,000$ iterations, and then it was reduced to 3×10^{-5} until reaching iteration $1,200,000$. The learning rate was further reduced to 3×10^{-6} until reaching $3,000,000$ iteration. The training was done on 4 GPUs. This resulted in WW-Nets that produced comparable or better performance than state-of-the-art methods. Performance results for PASCAL VOC 2012 are shown in Table II.

COCO Results. We also measured WW-Nets performance on the COCO dataset. The What Net was VGG16, with features for calculating GT extracted from the last convolutional layer. The Where Net was Resnet 101. Initialized with parameters previously learned for the PASCAL VOC 2012 evaluation, the network was further trained using the COCO training-dev set. The training was done on 4 GPUs. This resulted in WW-Nets that produced comparable or better performance than state-of-the-art methods, often by a large margin. Performance results for COCO are shown in Table III.

V. HUMAN ATTENTION

Participants. 15 healthy undergraduate students (12 female, 3 male; age: mean$\pm$s.d.$= 20.06 \pm 1.62$) were recruited from the University of California, Merced campus. Participants provided informed consent in accordance with IRB protocols and received one hour of course credit for their participation. Participation was restricted to those who reported normal, uncorrected vision in a pre-screen survey.

Materials. The images seen were a subset of 200 images from PASCAL VOC 2007. For display, images were scaled up to double their original size. The display width of images ranged from $636 - 1000$ pixels (mean$= 953.68$; median$=$

TABLE III
COMPARISON ON COCO TEST-DEV REVEALS THAT WW-NET PERFORMS FAVORABLY AGAINST STATE-OF-THE-ART METHODS.

Method	Backbone	Input Resolution	AP	AP_{50}	AP_{75}
Faster R-CNN w/ FPN [38]	ResNet-101	1000×600	49.5	59.1	39.0
Deformable-CNN [39]	Inception-ResNet	1000×600	-	58.0	-
Deep Regionlets [40]	ResNet-101	1000×600	-	59.8	-
YOLOv2 [14]	DarkNet-19	544×544	31.6	44.0	19.2
YOLOv3 [41]	DarkNet-53	608×608	46.1	57.9	34.4
SSD [16]	ResNet-101	513×513	41.8	50.4	33.3
DSSD [33]	ResNet-101	513×513	44.2	53.3	35.2
RetinaNet [42]	ResNet-101	1333×800	50.7	59.1	42.3
WW-Net (ours)	Resnet 101	511×511	51.6	57.8	45.3

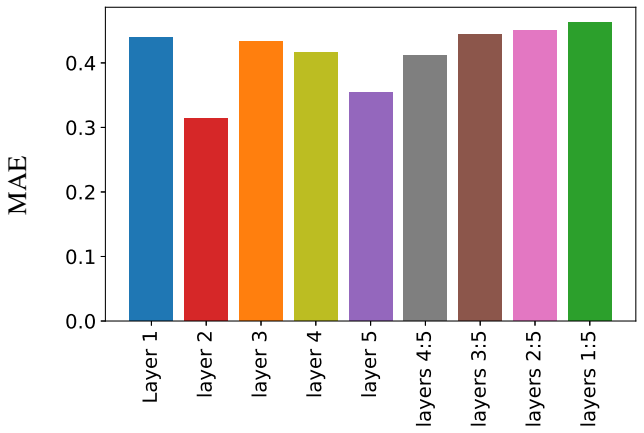

Fig. 3. MAE Measure Between Human Fixation Distributions and Attention Maps at Different Layers of WW-Nets

1000; standard deviation=102.54). The display height of images ranged from $350-1000$ pixels (mean$=$ 759.71; median$=$ 750; standard deviation$=$ 120.73). The ordering of images was randomized within participants.

Procedure. Participants completed the task individually in a research laboratory. Participants were seated at a desk in front of a computer and were fitted with a head-mounted Eyelink II eye tracker system. A microphone was placed nearby to record participants' speech. Prior to the beginning of the experiment, the eye-tracker was calibrated using a standard nine-point grid, and the subject was shown how to perform a drift correction, which took place at the beginning of each trial. Eye movement data was collected via the Eyelink control software and custom MATLAB scripts. Data from the right eye were collected using both pupil shape and corneal reflection. Each trial began with the participant fixating the center of the screen and pressing the space bar to initiate the trial. Then, an image was displayed in the center of the screen for 5 seconds. Participants were instructed to name out loud as many different objects as they could identify in the image, within a 5-second time limit.

Fixation Heat-Map. The raw eye-tracking data were converted into MatLab data structures using the `Edf2Mat` package. Heat maps were generated from the eye-fixation data. Fixations were included in the analysis only if they began at or after the start of the trial. Fixations were pooled from all participants. For each image, a zero matrix with the same $N \times M$ dimensions as the original image pixel height and width was created. Fixation coordinates were scaled so that they corresponded to locations within the original image sizes and then rounded to the nearest integer. Thus, the coordinate values for each fixation corresponded to a location within the matrix. For each fixation, the value of the corresponding matrix position was increased by the duration of the fixation in milliseconds. Then, all values of the matrix were divided by the maximum value of the matrix in order to normalize all matrix values to the $[0, 1]$ range. Finally, a convolution was performed on the matrix using a Gaussian kernel ($\sigma = 20$, size = 80×80). These steps yielded a heat map showing the likely places to which participants attend within the images. Calculations were performed in MATLAB using custom scripts.

Eye-Tracking Study Results. Example distribution heatmaps of human fixations are shown in Figure 2. We used these distributions as a form of ground-truth for analyzing the attention maps produced by WW-Nets. We compared the attention map distributions, $G(x, y)$, with the human fixation distributions, $S(x, y)$, using a simple Mean Absolute Error (MAE) measure.

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x, y) - G(x, y)| \qquad (11)$$

where W and H are the width and height of the image. We performed this assessment for various attention maps in the network. The resulting error values are shown in Figure 3.

We found that the second convolutional layer displayed the greatest similarity to human performance. This contrasts with the attention mechanisms in other object detection frameworks, which frequently base attention only on the last convolutional layer. It appears as if the high resolution and fundamental features learned by the network provide relatively good guides for attention during object detection. Both the object detection system software and the fixation distribution data collected from human subjects will be made publicly available.

VI. CONCLUSION

In this paper, we highlighted the utility of incorporating selective attention mechanisms into object detection algorithms. We suggested that such mechanisms could guide the

search over image regions, focusing this search in an informed manner. In addition, we demonstrated that the resulting removal of distracting irrelevant material can improve object detection accuracy substantially. Our approach was inspired by the visual system of the human brain. Theories of spatial attention that see it as arising from dual interacting "what" and "where" visual streams led us to propose a dual network architecture for object detection. The resulting architecture, WW-Nets, integrates attention based object detection methods with supervised approaches.

The benefits of selective attention, as implemented in WW-Nets, are evident in the performance results reported on the PASCAL VOC 2007, PASCAL VOC 2012, and COCO datasets. Evaluation experiments revealed that WW-Nets display greater object detection accuracy than state-of-the-art approaches, often by a large margin.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, vol. 7, 2017.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, 2017.

[5] W. Ge, S. Yang, and Y. Yu, "Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning," in *CVPR*, 2018.

[6] X. Wang, S. You, X. Li, and H. Ma, "Weakly-supervised semantic segmentation by iteratively mining common object features," in *CVPR*, 2018.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *CVPR*, 2017.

[8] F. Chen, R. Ji, X. Sun, Y. Wu, and J. Su, "Groupcap: Group-based image captioning with structured relevance and diversity constraints," in *CVPR*, 2018.

[9] J. Aneja, A. Deshpande, and A. Schwing, "Convolutional image captioning," in *CVPR*, 2018.

[10] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *ICML*, 2015.

[11] R. Girshick, "Fast R-CNN," in *CVPR*, 2015.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[14] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[17] J. Moran and R. Desimone, "Selective attention gates visual processing in the extrastriate cortex," *Science*, 1985.

[18] M. K. Ebrahimpour, J. Li, Y.-Y. Yu, J. Reesee, A. Moghtaderi, M.-H. Yang, and D. C. Noelle, "Ventral-dorsal neural networks:object detection via selective attention," in *WACV*, 2019.

[19] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *CVPR*, 2016.

[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *See https://arxiv.org/abs/1610.02391 v3*, vol. 7, no. 8, 2016.

[21] P. Zhou, B. Ni, C. Geng, J. Hu, and Y. Xu, "Scale-transferrable object detection," in *CVPR*, 2018.

[22] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *CVPR*, 2018.

[23] Y. Zhang, Y. Bai, M. Ding, Y. Li, and B. Ghanem, "W2f: A weakly-supervised to fully-supervised framework for object detection," in *CVPR*, 2018.

[24] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *IJCV*, vol. 104, no. 2, pp. 154–171, 2013.

[25] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection." in *CVPR*, 2017.

[26] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[27] ——, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[29] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *CVPR*, 2016.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[31] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *CVPR*, 2015.

[32] K. J. Dai and Y. L. R-FCN, "Object detection via region-based fully convolutional networks. arxiv preprint," *arXiv preprint arXiv:1605.06409*, 2016.

[33] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[34] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *CVPR*, 2016.

[35] S. Gidaris and N. Komodakis, "Object detection via a multi-region and semantic segmentation-aware cnn model," in *CVPR*, 2015.

[36] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," *PAMI*, vol. 39, no. 7, pp. 1476–1481, 2017.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009.

[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.

[39] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *ICCV*, 2017.

[40] H. Xu, X. Lv, X. Wang, Z. Ren, N. Bodla, and R. Chellappa, "Deep regionlets for object detection," in *ECCV*, 2018.

[41] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.