Towards Universal Object Detection by Domain Attention

Xudong Wang[1], Zhaowei Cai[1], Dashan Gao[2] and Nuno Vasconcelos[1]
[1]University of California, San Diego, [2]12 Sigma Technologies
{xuw080,zwcai,nuno}@ucsd.edu, dgao@12sigma.ai

Abstract

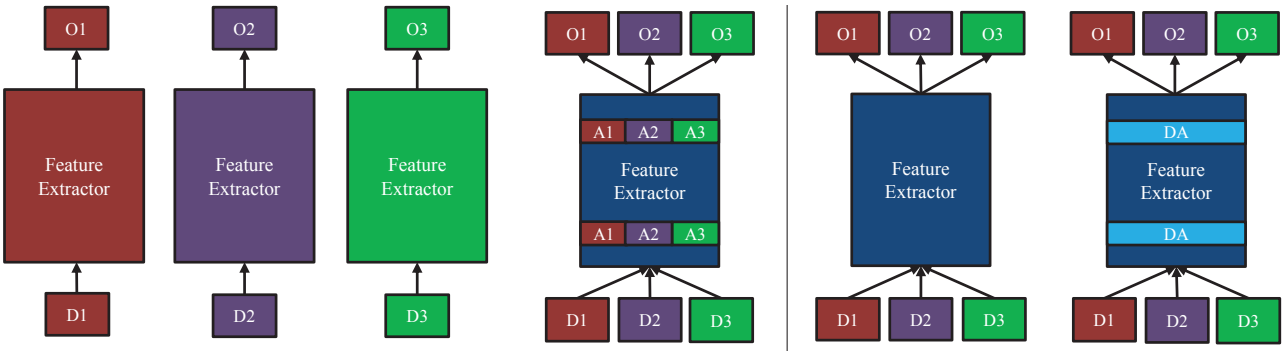
Despite increasing efforts on universal representations for visual recognition, few have addressed object detection. In this paper, we develop an effective and efficient universal object detection system that is capable of working on various image domains, from human faces and traffic signs to medical CT images. Unlike multi-domain models, this universal model does not require prior knowledge of the domain of interest. This is achieved by the introduction of a new family of adaptation layers, based on the principles of squeeze and excitation, and a new domain-attention mechanism. In the proposed universal detector, all parameters and computations are shared across domains, and a single network processes all domains all the time. Experiments, on a newly established universal object detection benchmark of 11 diverse datasets, show that the proposed detector outperforms a bank of individual detectors, a multi-domain detector, and a baseline universal detector, with a 1.3× parameter increase over a single-domain baseline detector. The code and benchmark are available at http://www.svcl.ucsd.edu/projects/universal-detection/.

1. Introduction

There has been significant progress in object detection in recent years [11, 44, 2, 26, 13, 3], powered by the availability of challenging and diverse object detection datasets, e.g. PASCAL VOC [6], COCO [27], KITTI [9], WiderFace [58], etc. However, existing detectors are usually *domain-specific*, e.g. trained and tested on a single dataset. This is partly due to the fact that object detection datasets are diverse and there is a nontrivial domain shift between them. As shown in Figure 1, detection tasks can vary in terms of categories (human face, horse, medical lesion, etc.), camera viewpoints (images taken from aircrafts, autonomous vehicles, etc.), image styles (comic, clipart, watercolor, medical), etc. In general, high detection performance requires a detector specialized on the target dataset.

This poses a significant problem for practical applications, which are not usually restricted to any one of the domains of Figure 1. Hence, there is a need for systems capable of detecting objects regardless of the domain in which images are collected. A simple solution is to design a *specialized* detector for each domain of interest, e.g. use D detectors trained on D datasets, and load the detector specialized to the domain of interest at each point in time. This, however, may be impractical, for two reasons. First, in most applications involving autonomous systems the domain of interest can change frequently and is not necessarily known a priori. Second, the overall model size increases linearly with the number of domains D. A recent trend, known as general AI, is to request that a single universal model solves multiple tasks [21, 25, 62], or the same task over multiple domains [40, 1]. However, existing efforts in this area mostly address image classification, rarely targeting the problem of object detection. The fact that modern object detectors are complex systems, composed of a backbone network, proposal generator, bounding box regressor, classifier, etc., makes the design of a universal object detec-

Figure 1. Samples of our universal object detection benchmark.



(a) Single-domain Detector Bank (b) Adaptive Multi-domain Detector (c) Universal Detector (d) Domain-attentive Universal Detector

Figure 2. Multi-domain and universal object detectors for three domains. "D" is the domain, "O" the output, "A" domain-specific adapter, and "DA" the proposed domain attention module. The blue color and the DA are domain-universal, but the other colors domain-specific.

tor much more challenging than a universal image classifier.

In this work, we consider the design of an object detector capable of operating over multiple domains. We begin by establishing a new universal object detection benchmark, denoted as UODB, consisting of 11 diverse object detection datasets (see Figure 1). This is significantly more challenging than the Decathlon [40] benchmark for multi-domain recognition. To the best of our knowledge, we are the first to attack universal object detection using deep learning. We expect this new benchmark will encourage more efforts in the area. We then propose a number of architectures, shown in Figure 2, to address the universal/multi-domain detection problem.

The two architecture on the left of Figure 2 are multi-domain detectors, which require prior knowledge of the domain of interest. The two architectures on the right are universal detectors, with no need for such knowledge. When operating on an unknown domain, the multi-domain detector have to repeat the inference process with different sets of domain-specific parameters, while the universal detector performs inference only once. The detector of Figure 2 (a) is a bank of domain-specific detectors, with no sharing of parameters/computations. Multi-domain learning (MDL) [20, 35, 24, 59, 19, 5] improves on this, by sharing parameters across various domains, and adding small domain-specific layers. In [40, 1], expensive convolutional layers are shared and complemented with light-weight domain-specific *adaptation layers*. Inspired by these, we propose a new class of light adapters for detection, based on the squeeze and excitation (SE) mechanism of [15], and denoted *SE adapters*. This leads to the *multi-domain detector* of Figure 2 (b), where domain-specific SE adapters are introduced throughout the network to compensate for domain shift. On UODB, this detector outperforms that of Figure 2 (a) with ∼5 times fewer parameters.

In contrast, the *universal detector* of Figure 2 (c) shares all parameters/computations (other than output layers) across domains. It consists of a *single* network, which is always active. This is the most efficient solution in terms of parameter sharing, but it is difficult for a single model to cover many domains with nontrivial domain shifts. Hence, this solution underperforms the multi-domain detector of Figure 2 (b). To overcome this problem, we propose the *domain-attentive universal detector* of Figure 2 (d). This leverages a novel domain attention (DA) module, in which a bank of the new universal SE adapters (active at all times) is first added, and a feature-based attention mechanism is then introduced to achieve domain sensitivity. This module learns to assign network activations to different domains, through the universal SE adapter bank, and soft-routs their responses by the domain-attention mechanism. This enables the adapters to specialize on individual domains. Since the process is data-driven, the number of domains does not have to match the number of datasets and datasets can span multiple domains. This allows the network to leverage shared knowledge across domains, which is not available in the common single-domain detectors. Our experiments, on the newly established UODB, show that this data-driven form of parameter/computation sharing enables substantially better multi-domain detection performance than the remaining architectures of Figure 2.

2. Related Work

Object Detection: The two stage detection framework of the R-CNN [12], Fast R-CNN [11] and Faster R-CNN [44] detectors has achieved great success in recent years. Many works have expanded this base architecture. For example, MS-CNN [2] and FPN [26] built a feature pyramid to effectively detect objects of various scales; the R-FCN [4] proposed a position-sensitive pooling to achieve further speed-ups; and the Cascade R-CNN [3] introduced a multi-stage cascade for high quality object detection. In parallel, single-stage object detectors, such as YOLO [42] and SSD [29], became popular for their fairly good performance and high

speed. However, none of these detectors could reach high detection performance on more than one dataset/domain without finetuning. In the pre-deep learning era, [23] proposed a universal DPM [8] detector, by adding dataset specific biases to the DPM. But this solution is limited since DPM is not comparable to deep learning detectors.

Multi-Task Learning: Multi-task learning (MTL) investigates how to jointly learn multiple tasks simultaneously, assuming a single input domain. Various multi-task networks [25, 62, 13, 28, 50, 63] have been proposed for joint solution of tasks such as object recognition, object detection, segmentation, edge detection, human pose, depth, action recognition, etc., by leveraging information sharing across tasks. However, the sharing is not always beneficial, sometimes hurting performance [7, 22]. To address this, [32] proposed a cross-stitch unit, which combines tasks of different types, eliminating the need to search through several architectures on a per task basis. [62] studied the common structure and relationships of several different tasks.

Multi-Domain Learning/Adaptation: Multi-domain learning (MDL) addresses the learning of representations for multiple domains, known a priori [20, 36]. It uses a combination of parameters that are shared across domains and domain-specific parameters. The latter are adaptation parameters, inspired by works on domain adaptation [38, 30, 46, 31], where a model learned from a source domain is adapted to a target domain. [1] showed that multi-domain learning is feasible by simply adding domain-specific BN layers to an otherwise shared network. [40] learned multiple visual domains with residual adapters, while [41] empirically studied efficient parameterizations. However, they build on BN layers and are not suitable for detection, due to the batch constraints of detector training. Instead, we propose an alternative SE adapters, inspired by "Squeeze-and-Excitation" [15], to solve this problem.

Attention Module: [49] proposed a self-attention module for machine translation, and similarly, [51] proposed a non-local network for video classification, based on a spacetime dependency/attention mechanism. [15] focused on channel relationships, introducing the SE module to adaptatively recalibrate channel-wise feature responses, which achieved good results on ImageNet recognition. In this work, we introduce a domain attention module inspired by SE to make data-driven domain assignments of network activations, for the more challenging problem of universal object detection.

3. Multi-domain Object Detection

The problem of multi-domain object detection is to detect objects on various domains.

3.1. Universal Object Detection Benchmark

To train and evaluate universal/multi-domain object detection systems, we established a new universal object detection benchmark (UODB) of 11 datasets: Pascal VOC [6], WiderFace [58], KITTI [9], LISA [33], DOTA [53], COCO [27], Watercolor [17], Clipart [17], Comic [17], Kitchen [10] and DeepLesions [55]. This set includes the popular VOC [6] and COCO [27], composed of images of everyday objects, e.g. bikes, humans, animals, etc. The 20 VOC categories are replicated on CrossDomain [17] with three subsets of Watercolor, Clipart and Comic, with objects depicted in watercolor, clipart and comic styles, respectively. Kitchen [10] consists of common kitchen objects, collected with an hand-held Kinect, while WiderFace [58] contains human faces, collected on the web. Both KITTI [9] and LISA [33] depict traffic scenes, collected with cameras mounted on moving vehicles. KITTI covers the categories of vehicle, pedestrian and cyclist, while LISA is composed of traffic signs. DOTA [53] is a surveillance-style dataset, containing objects such as vehicles, planes, ships, harbors, etc. imaged from aerial cameras. Finally DeepLesion [55] is a dataset of lesions on medical CT images. A representative example of each dataset is shown in Figure 1. Some more details are summarized in Table 1. Altogether, UODB covers a wide range of variations in category, camera view, image style, etc, and thus establishes a good suite for the evaluation of universal/multi-domain object detection.

3.2. Single-domain Detector Bank

The Faster R-CNN [44] is used as the baseline architecture of all detectors proposed in this work. As a single-domain object detector, the Faster R-CNN is implemented in two stages. First, a region proposal network (RPN) produces preliminary class-agnostic detection hypotheses. The second stage processes these with a region-of-interest detection network to output the final detections.

As illustrated in Figure 2 (a), the simplest solution to multi-domain detection is to use an independent detector per dataset. We use this detector bank as a multi-domain detection baseline. This solution is the most expensive, since it implies replicating all parameters of all detectors. Figure 3 shows the statistics (mean and variance) of the convolutional activations of the 11 detectors on the corresponding dataset. Some observations can be made. First, these statistics vary non-trivially across datasets. While the activation distributions of VOC and COCO are similar, DOTA, DeepLesion and CrossDomain have relatively different distributions. Second, the statistics vary across network layers. Early layers, which are more responsible for correcting domain shift, have more evident differences than latter layers. This tends to hold up to the output layers. These are responsible for the assignment of images to different categories and naturally differ. Interestingly, this behavior also holds for RPN layers, even though they are category-independent. Third, many layers have similar statistics across datasets. This is especially true for intermediate layers, suggesting

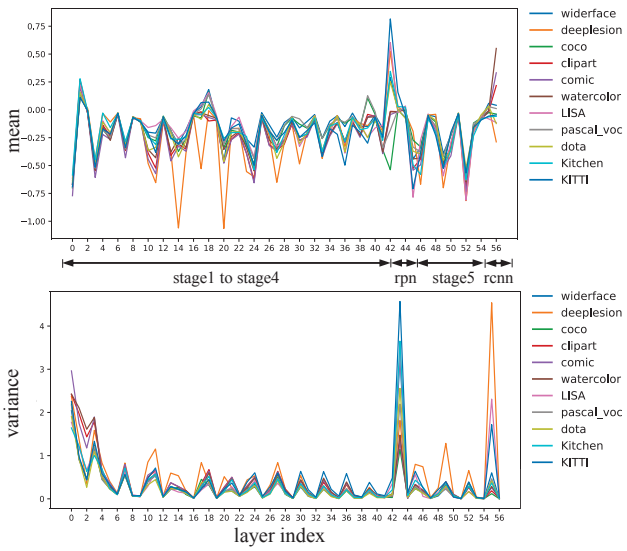


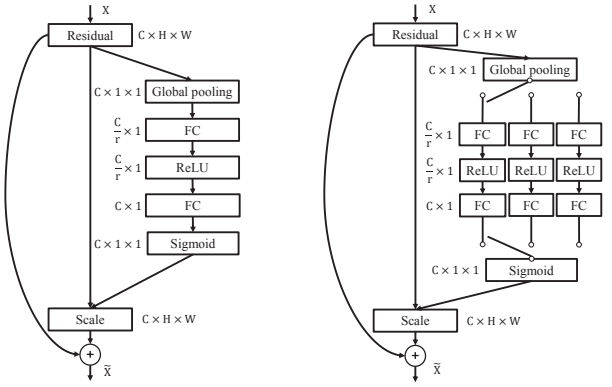
Figure 3. The activation statistics of all single-domain detectors.

that they can be shared by at least some domains.

3.3. Adaptive Multi-domain Detector

Inspired by Figure 3, we propose an adaptive multi-domain detector, shown in Figure 2 (b). In this model, the output and RPN layers are domain-specific. The remainder of the network, e.g. all convolutional layers, is shared. However, to allow adaptation to new domains, we introduce some additional domain-specific layers, as is commonly done in MDL [40, 1]. These extra layers should be 1) sufficiently powerful to compensate for domain shift; 2) as light as possible to minimize parameters/computation. The adaptation layers of [40, 1] rely extensively on BN. This is unfeasible for detection, where BN layers have to be frozen, due to the small batch sizes allowable for detector training.

Instead, we have experimented with the squeeze-and-excitation (SE) module [15] of Figure 4 (a). There are a few reasons for this. First, feature-based attention is well known to be used in mammalian vision as a mechanism to adapt perception to different tasks and environments [61, 37, 52, 18, 60]. Hence, it seems natural to consider feature-based attention mechanisms for domain adaptation. Second, the SE is a module that accounts for interdependencies among channels to modulate channel responses. This can be seen as a a feature-based attention mechanism. Third the SE module has enabled the SENet to achieve state-of-the-art classification on ImageNet. Finally, it is a light-weight module. Even when added to each residual block of the ResNet [14] it increases the total parameter count by only $\sim$10%. This is close to what was reported by [40] for BN-based adapters. For all these reasons, we adopt the SE module as the atomic adaptation unit, used to build all domain adaptive detectors proposed in this work, and denote it by the *SE adapter*.



(a) SE adapter (b) SE adapter bank

Figure 4. (a) block diagram of SE adapter and (b) SE adapter bank.

3.4. SE Adapters

Following [15], the *SE adapter* consists of the sequence of operations of Figure 4 (a): a global pooling layer, a fully connected (FC) layer, a ReLU layer, and a second FC layer, implementing the computation

$$\mathbf{X}_{SE} = \mathbf{F}_{SE}(\mathbf{F}_{avg}(\mathbf{X})), \tag{1}$$

where $\mathbf{F}_{avg}$ is a global average pooling operator, and $\mathbf{F}_{SE}$ the combination of FC+ReLU+FC layers. The channel dimension reduction factor r, in Figure 4, is set as 16 in our experiments. To enable multi-domain object detection, the SE adapter is generalized to the architecture of Figure 4 (b), which is denoted as the *SE adapter bank*. This consists of adding a SE adapter branch per domain and a domain-switch, which allows the selection of the SE adapter associated with the domain of interest. Note that this architecture assumes this domain to be known a priori. It leads to the *multi-domain detector* of Figure 2 (b). Compared to Figure 2 (a), this model is up to 5 times smaller, while achieving better overall performance across the 11 datasets.

4. Universal Object detection

The detectors of the previous section require prior knowledge of the domain of interest. This is undesirable for autonomous systems, like robots or self-driving cars, where determining the domain is part of the problem to solve. In this section, we consider the design of *universal detectors,* which eliminate this problem.

4.1. Universal Detector

The simplest solution to universal detection, shown in Figure 2 (c), is to share a single detector by all tasks. Note that, even for this detector, the output layer has to be task-specific, by definition of the detection problem. We have found that there is also a benefit in using task-specific RPN layers, due to the observations of Figure 3. This is not a

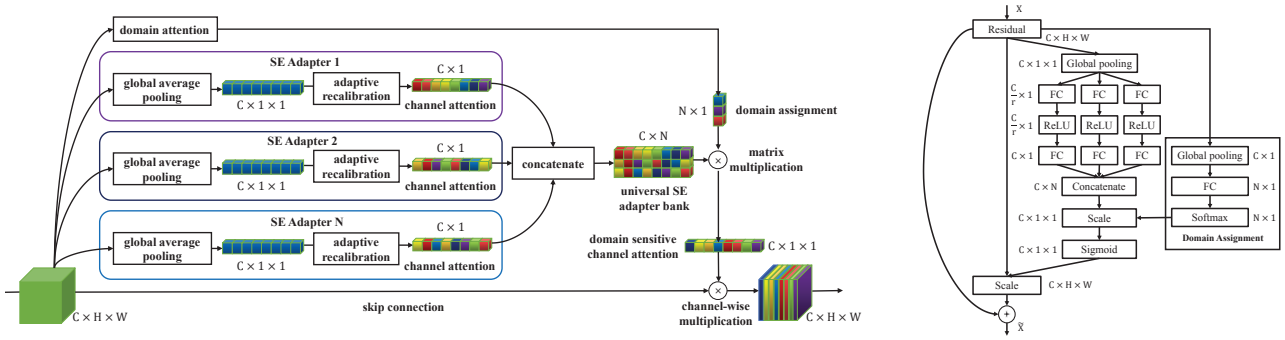


Figure 5. The block diagram (left) and the detailed view (right) of the proposed domain adaptation module.

problem because the task, namely what classes the system is trying to detect, is always known. Universality refers to the domain of input images that the detector processes, which does not have to be known in the case of Figure 2 (c). Beyond universal, the fully shared detector is the most efficient of all detectors considered in this work, as it has no domain-specific parameters. On the other hand, by forcing the same set of parameters/representations on all domains, it has little flexibility to deal with the statistical variations of Figure 3. In our experiments, this detector usually underperforms the multi-domain detectors of Figure 2 (a) and (b).

4.2. Domain-attentive Universal Detector

Ideally, a universal detector should have some domain sensitivity, and be able to adapt to different domains. While this has a lot in common with multi-domain detection, there are two main differences. First, the domain must be inferred automatically. Second, there is no need to tie domains and tasks. For example, the traffic tasks of Figure 1 operate on a common visual domain, "traffic scenes", which can have many sub-domains, e.g. due to weather conditions (sunny vs. rainy), environment (city vs. rural), etc. Depending on the specific operating conditions, any of the tasks may have to be solved in any of the domains. In fact, the domains may not even have clear semantics, i.e. they can be data-driven. In this case, there is no need to request that each detector operates on a single domain, and a soft domain-assignment makes more sense. Given all of this, while domain adaptation can still be implemented with the SE adapter of Figure 4 (a), the hard attention mechanism of Figure 4 (b), which forces the network to fully attend to a single domain, can be suboptimal. To address this limitations, we propose the domain adaptation (DA) module of Figure 5. This has two components, a *universal SE adapter bank* and a *domain attention* mechanism, which are discussed next.

4.3. Universal SE Adapter Bank

The universal SE (USE) Adapter Bank, shown in Figure 5, is an SE adapter bank similar to that of Figure 4 (b). The main difference is that there is no domain switching, i.e. the adapter bank is *universal.* This is implemented by concatenating the outputs of the individual domain adapters to form a universal representation space

$$\mathbf{X}_{USE} = [\mathbf{X}_{SE}^1, \mathbf{X}_{SE}^2, ..., \mathbf{X}_{SE}^N] \in \mathbb{R}^{C \times N}, \qquad (2)$$

where N is the number of adapters and $\mathbf{X}_{SE}^i$ the output of each adapter, given by (1). Note that N is not necessarily identical to the number of detection tasks. The USE adapter bank can be seen as a non-linear generalization of the filter banks commonly used in signal processing [48]. Each branch (non-linearly) projects the input along a subspace matched to the statistics of a particular domain. The attention component then produces a domain-sensitive set of weights that are used to combine these projections in a data-driven way. In this case, there is no need to know the operating domain in advance. In fact there may not even be a single domain, since an input image can excite multiple SE adapter branches.

4.4. Domain Attention

The attention component, of Figure 5, produces a domain-sensitive set of weights that are used to combine the SE bank projections. Motivated by the SE module, the domain attention component first applies a global pooling to the input feature map, to remove spatial dimensions, and then a softmax layer (linear layer plus softmax function)

$$\mathbf{S}_{DA} = \mathbf{F}_{DA}(\mathbf{X}) = \text{softmax}(\mathbf{W}_{DA}\mathbf{F}_{avg}(\mathbf{X})), \qquad (3)$$

where $\mathbf{W}_{DA} \in \mathbb{R}^{N \times C}$ is the matrix of softmax layer weights. The vector $\mathbf{S}_{DA}$ is then used to weigh the USE bank output $\mathbf{X}_{USE}$, to produce a vector of domain adaptive responses

$$\mathbf{X}_{DA} = \mathbf{X}_{USE}\mathbf{S}_{DA} \in \mathbb{R}^{C \times 1}. \qquad (4)$$

As in the SE module of [15], $\mathbf{X}_{DA}$ is finally used to channel-wise rescale the activations $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ being adapted,

$$\widetilde{\mathbf{X}} = \mathbf{F}_{scale}(\mathbf{X}, \sigma(\mathbf{X}_{DA})) \qquad (5)$$

where $\mathbf{F}_{scale}(\cdot)$ implements a channel-wise multiplication, and σ is the sigmoid function.

dataset	dataset details			hyperparameters				mAP
	class	T/V/T	domain	size	BS	RoIs	S/R	
KITTI	3	7k/-/7k	traffic	576	256	128	12/3	64.3
WiderFace	1	13k/3k/16k	face	800	256	256	12/1	48.9
VOC	20	8k/8k/5k	natural	600	256	256	4/3	78.5
LISA	4	8k/-/2k	traffic	800	64	32	4/3	88.3
DOTA	15	14k/5k/10k	aerial	600	128	128	12/3	57.5
COCO	80	35k/5k/-	natural	800	256	256	4/3	47.3
Watercolor	6	1k/-/1k	watercolor	600	256	256	4/3	52.4
Clipart	6	0.5k/-/0.5k	clipart	600	256	256	4/3	32.1
Comic	20	1k/-/1k	comic	600	256	256	4/3	45.8
Kitchen	11	5k/-/2k	indoor	800	256	256	12/3	87.7
DeepLesion	1	23k/5k/5k	medical	512	128	64	12/3	51.3
Average	-	-	-	-	-	-	-	59.4

Table 1. The dataset details, the domain-specific hyperparameters and the performance of the single-domain detectors. "T/V/T" means train/val/test, "size" the shortest side of inputs, BS RPN batch size, and S/R anchor "scales/aspect ratios".

In this way, the USE bank captures the feature subspaces of the domains spanned by all datasets, and the DA mechanism soft-routes the USE projections. Both operations are data-driven, and operate with no prior knowledge of the domain. Unlike the hard attention mechanism of Figure 4 (b), this DA module enables information sharing across domains, leading to a more effective representation. In our experiments, the domain-attentive universal detector outperforms the other detectors of Figure 2.

5. Experiments

In all experiments, we used a PyTorch implementation [57] of the Faster R-CNN with the SE-ResNet-50 [15] pre-trained on ImageNet, as the backbone for all detectors. Training started with a learning rate of 0.01 for 10 epochs and 0.001 for another 2 epochs on 8 synchronized GPUs, each holding 2 images per iteration. All samples of a batch are from a single (randomly sampled) dataset, and in each epoch, all samples of each dataset are processed only once. As is common for detection, the first convolutional layer, the first residual block and all BN layers are frozen, during training. These settings were used in all experiments, unless otherwise noted. Both multi-domain and universal detectors were trained on all domains of interest simultaneously.

The Faster R-CNN has many hyperparameters. In the literature, where detectors are tested on a single domain, these are tuned to the target dataset, for best performance. This is difficult, and very tedious, to do over the 11 datasets now considered. We use the same hyperparameters across datasets, except when this is critical for performance and relatively easy to do, e.g. the choice of anchors. The main dataset-specific hyperparameters are shown in Table 1.

5.1. Datasets and Evaluation

Our experiments used the new UODB benchmark introduced in Section 3.1. For Watercolor [17], Clipart [17], Comic [17], Kitchen [10] and DeepLesion [55], we trained on the official `trainval` sets and tested on the `test` set. For Pascal VOC [6], we trained on VOC2007 and VOC2012 `trainval` set and tested on VOC2007 `test` set. For WiderFace [58], we trained on the `train` set and tested on the `val` set. For KITTI [9], we followed the train/val splitting of [2] for development and trained on the `trainval` set for the final results on `test` set. For LISA [33], we trained on the `train` set and tested on the `val` set. For DOTA [53], we followed the pre-processing of [53], trained on `train` set and tested on `val` set. For MS-COCO [27], we trained on COCO 2014 `valminusminival` and tested on `minival`, to shorten the experimental period.

All detectors were evaluated on each dataset individually. The Pascal VOC mean average precision (mAP) was used for evaluation in all cases. The average mAPs was used as the overall measure of universal/multi-domain detection performance. The domain attentive universal detector was also evaluated using the official evaluation tool of each dataset, for comparison with the literature.

	Params	time	KITTI	VOC	WiderFace	LISA	Kitchen	Avg
single-domain	31.06M×5	5x	64.3	78.5	48.8	88.3	87.7	73.5
adaptive	42.37M	6x	67.8	78.9	49.9	**88.5**	86.0	74.2
BNA [1]	31.72M	5x	64.0	71.9	44.0	66.8	84.3	66.2
RA [40]	82.72M	6x	64.3	70.5	46.9	69.1	84.6	67.1
universal	31.64M	1x	66.3	76.7	45.5	88.4	85.4	72.5
universal+DA†	42.37M	1.3x	67.5	79.0	49.8	88.2	88.0	74.6
universal+DA	42.44M	1.33x	**67.9**	**79.2**	**52.2**	87.5	**88.5**	**75.1**

Table 2. The comparison on multi-domain detection. † denotes fixed assignment. "time" is the relatively run-times on the five datasets when the domain is unknown.

5.2. Single-domain Detection

Table 1 shows the results of the single-domain detector bank of Figure 2 (a) on all datasets. Our VOC baseline with the SE-ResNet-50 is 78.5, and better than the Faster R-CNN performance of [45, 14] (76.4 mAP for ResNet-101). The other entries in the table are incomparable to the literature, where different evaluation metrics/tools are used for different datasets. The detector bank is a fairly strong baseline for multi-domain detection (average mAP of 59.4).

5.3. Multi-domain Detection

Table 2 compares the multi-domain object detection performance of all architectures of Figure 2. For simplicity, only five datasets (VOC, KITTI, WiderFace, LISA and Kitchen) were used in this section. The table confirms that the adaptive multi-domain detector of Section 3.3 ("adaptive") is light-weight, only adding ∼11M parameters to the Faster R-CNN over the five datasets. Nevertheless, it outperforms the much more expensive single-domain detector bank by 0.7 points. Note that the latter is a strong baseline, showing the multi-domain detector can beat individually trained models with a fraction of the computation. Ta-

	# adapters	Params	DA index	KITTI	VOC	WiderFace	LISA	Kitchen	COCO	DOTA	DeepLesion	Comic	Clipart	Watercolor	Avg
single-domain	-	31.06M×11	-	64.3	78.5	48.8	88.3	87.7	**47.3**	**57.5**	51.2	45.8	32.1	52.6	59.4
universal	-	32.60M	-	67.5	80.9	45.5	87.1	88.5	45.5	54.7	45.3	51.1	43.1	47.0	59.7
adaptive	11	58.13M	-	68.0	82.1	50.6	**88.5**	87.2	45.7	54.1	53.0	50.0	56.1	57.8	63.0
universal+DA	11	58.29M	all	**68.1**	82.0	51.6	88.3	**90.1**	46.5	57.0	**57.3**	50.7	53.1	**58.4**	63.8
universal+DA*	6	41.74M	first+middle	67.6	**82.7**	**51.8**	87.9	88.7	46.8	57.0	54.8	**52.6**	**54.6**	58.2	**63.9**

Table 3. Overall results on the full universal object detection benchmark (11 datasets).

# adapters	Params	KITTI	VOC	WiderFace	LISA	Kitchen	Avg
single	31.06M×5	64.3	78.5	48.8	88.3	87.7	73.5
1	32.32M	66.3	74.9	43.5	87.4	85.4	71.3
3	37.38M	67.8	78.4	47.1	87.7	**89.0**	74.1
5	42.44M	**67.9**	79.2	**52.2**	87.5	88.5	75.1
7	47.50M	**67.9**	**79.6**	**52.2**	**89.5**	88.7	**75.6**

Table 4. The effect of SE adapters number.

ble 2 also shows that the proposed SE adapter significantly outperforms the BN adapter (BNA) of [1] and the residual adapter (RA) or [40], previously proposed for classification. This is not surprising, given the above discussed inadequacy of BN as an adaptation mechanism for object detection.

The universal detector of Figure 2 (c) is even more efficient, adding only 0.5M parameters to the Faster R-CNN, accounting for domain-specific RPN and output layers. However, its performance ("universal" in Table 2) is much weaker than that of the adaptive multi-domain detector (1.7 points). Finally, the domain-attentive universal detector ("universal+DA") has the best performance. With a ∼7% parameter increase per domain, i.e. comparable to the multi-domain detector, it outperforms the single-domain bank baseline by 1.6 points. To assess the importance of data-driven domain attention mechanism of Figure 5 (b), we fixed the soft domain assignments, simply averaging the SE adapter responses, during both training and inference. This (denoted "universal+DA†") caused a performance drop of 0.5 point. Finally, Table 2 shows the relative run-times of all methods on the five datasets, when the domain is unknown. It can be seen that "universal+DA" is about 4× faster than the multi-domain detectors ("single-domain" and "adaptive") and only 1.33× slower than "universal".

5.4. Effect of the number of SE adapters

For the USE bank of Figure 5 (b), the number N of SE adapters does not have to match the number of detection tasks. Table 4 summarizes how the performance of the domain attentive universal detector depends on N. For simplicity, we again use 5 datasets in this experiment. For a single adapter, the DA module reduces to the standard SE module, and the domain attentive universal detector to the universal detector. This has the worst performance. Performance improves with the number of adapters. On the other hand, the number of parameters increases linearly with the number of adapters. In these experiments, the best trade-off between performance and parameters is around 5 adapters.

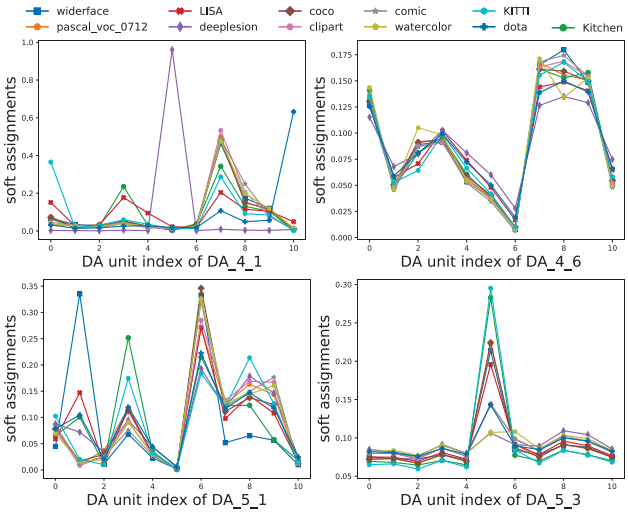


Figure 6. Soft assignments across SE units for all datasets.

This suggests that, while a good rule of thumb is to use "as many adapters as domains", fewer adapters can be used when complexity is at a premium.

5.5. Results on the full benchmark

Table 3 presents results on the full benchmark. The settings are as above, but we used 10 epochs with learning rate 0.1, and then 4 epochs with 0.01 on 8 GPUs, each holding 2 images. The universal detector performs comparably to the single-domain detector bank, with 10 times fewer parameters. The domain-attentive universal detector ("universal+DA") improves baseline performance by 4.4 points with a 5-fold parameter decrease. It has large performance gains (>5 points) on DeepLesion, Comic, and Clipart. This is because Comic/Clipart contain underpopulated classes, greatly benefiting from information leveraged from other domains. The large gain of DeepLesion is quite interesting, given the nontrivial domain shift between its medical CT images and the RGB images of the other datasets. The gains are mild for VOC, KITTI, Kitchen, WiderFace and WaterColor (1∼5 points), and none for COCO, LISA and DOTA. In contrast, for the universal detector, joint training is not always beneficial. This shows the importance of domain sensitivity for universal detection.

To investigate what was learned by the domain attention module of Figure 5 (b), we show the soft assignments of each dataset, averaged over its validation set, in Figure 6. Only the first and last blocks of the 4th and 5th residual

	Backbone	mAP
Faster-RCNN [44]	ResNet-101	76.4
R-FCN [4]	ResNet-50	77.0
Faster-RCNN‡ [45]	VGG16	78.8
Faster-RCNN (ours)	SE-ResNet-50	78.5
Faster-RCNN+DA	SE-ResNet-50	79.6
Faster-RCNN+DA†	SE-ResNet-50	82.7

(a) The comparison on VOC 2007 `test`. ‡/† denotes with COCO `trainval`/`val`.

	Backbone	Easy	Medium	Hard
Faster-RCNN [44]	VGG-16	0.907	0.850	0.492
MS-CNN [2]	VGG-16	0.916	0.903	0.802
HR [16]	ResNet-101	0.925	0.910	0.806
SSH [34]	VGG-16	0.931	0.921	0.845
Faster-RCNN (ours)	SE-ResNet-50	0.910	0.872	0.556
Faster-RCNN+DA	SE-ResNet-50	0.914	0.882	0.587

(b) The comparison on WiderFace `Val`.

	Backbone	Sensitivity
Faster-RCNN [44]	VGG-16	81.62
R-FCN [4]	VGG-16	82.21
3-DCE, 9 Slices [54]	VGG-16	84.34
3-DCE, 27 Slices [54]	VGG-16	85.65
Faster-RCNN (ours)	SE-ResNet-50	82.44
Faster-RCNN+DA	SE-ResNet-50	87.29

(c) Sensitivity at 4 FPs per image on DeepLesion `test` set.

	Backbone	Clipart	Watercolor	Comic
ADDA [47]	VGG-16	27.4	49.8	49.8
Faster-RCNN[44]	VGG-16	26.2	-	-
SSD300 [29]	VGG-16	26.8	49.6	24.9
Faster-RCNN+DT+PL[17]	VGG-16	34.9	-	-
SSD300+DT+PL[17]	VGG-16	46.0	54.3	37.2
Faster-RCNN (ours)	SE-ResNet-50	32.1	52.6	45.8
Faster-RCNN+DA	SE-ResNet-50	54.6	58.2	52.6

(d) The comparison on Clipart, Watercolor and Comic `test` set.

	Backbone	Moderate	Easy	Hard
Faster-RCNN [44]	VGG-16	81.84	86.71	71.12
SDP+CRC [56]	VGG-16	83.53	90.33	71.13
YOLOv3 [43]	Darknet-53	84.13	84.30	76.34
MS-CNN [2]	VGG-16	88.83	90.46	74.76
F-PointNet [39]	PointNet	90.00	90.78	80.80
Faster-RCNN (ours)	SE-ResNet-50	81.83	90.34	71.23
Faster-RCNN+DA	SE-ResNet-50	88.23	90.45	74.21

(e) The comparison on KITTI `test` set of car.

Table 5. The comparison with official evaluation on Pascal VOC, KITTI, DeepLesion, Clipart, Watercorlor, Comic and WiderFace.

stages are shown. The fact that some datasets, e.g. VOC and COCO, have very similar assignment distributions, suggests a substantial domain overlap. On the other hand, DOTA and DeepLesion have distributions quite distinct from the remaining. For example, on block "DA_4_1", DeepLesion fully occupies a single domain. These observations are consistent with Figure 3, indicating that the proposed DA module is able to learn domain-specific knowledge.

A comparison of the first and the last blocks of each residual stage, e.g. "DA_4_1" v.s. "DA_4_6", shows that the latter are much less domain sensitive than the former, suggesting that they could be made universal. To test this hypothesis, we trained a model with only 6 SE adapters for the 11 datasets, and only in the first and middle blocks, e.g. "DA_4_1" and "DA_4_3". This model, "universal+DA*", achieved the best performance with much less parameters than the "universal+DA" detector of 11 adapters. It outperformed the single domain baseline by 4.5 points.

5.6. Official evaluation

Since, to the best of our knowledge, this is the first work to explore universal/multi-domain object detection on 11 datasets, there is no literature for a direct comparison. Instead, we compared the "universal+DA*" detector of Table 3 to the literature using the official evaluation for each dataset. This is an unfair comparison, since the universal detector has to remember 11 tasks. On VOC, we trained two models, with/without COCO. Results are shown in Table 5a, where all methods were trained on Pascal VOC 07+12 `trainval`. Note that our Faster R-CNN baseline (SE-ResNet-50 backbone) is stronger than that of [14] (ResNet-101). Adding universal domain adapters improved on the baseline by more than 1.1 points. Adding COCO enabled another 3.1 points. Note that, 1) this universal training is different from the training scheme of [45] (the network trained on COCO then finetuned on VOC), where the final model is only optimized for VOC; and 2) only the 35k images of COCO2014 `valminusminival` were used.

The baseline was the default Faster R-CNN that initially worked on VOC, with minimum dataset-specific changes, e.g. in Table 1. Table 5e shows that this performed weakly on KITTI. However, the addition of adapters, enabled a gain of 6.4 points (`Moderate` setting). This is comparable to detectors optimized explicitly on KITTI, e.g. MS-CNN [2] and F-PointNet [39]. For WiderFace, which has enough training face instances, the gains of shared knowledge are smaller (see Table 5b). On the other hand, on DeepLesion and CrossDomain (Clipart, Comic and Watercolor), see Table 5c and 5d respectively, the domain attentive universal detector significantly outperformed the state-of-the-art. Overall, these results show that a single detector, which operates on 11 datasets, is competitive with single-domain detectors in highly researched datasets, such as VOC or KITTI, and substantially better than the state-of-the-art in less explored domains. This is achieved with a relatively minor increase in parameters, vastly smaller than that needed to deploy 11 single task detectors.

6. Conclusion

We have investigated the unexplored and challenging problem of universal/multi-domain object detection. We proposed a universal detector that requires no prior domain knowledge, consisting of a *single* network that is active for all tasks. The proposed detector achieves domain sensitivity through a novel data-driven domain adaptation module and was shown to outperform multiple universal/multi-domain detectors on a newly established benchmark, and even individual detectors optimized for a single task.

Acknowledgment This work was partially funded by NSF awards IIS-1546305 and IIS-1637941, a gift from 12 Sigma Technologies, and NVIDIA GPU donations.

References

[1] Hakan Bilen and Andrea Vedaldi. Universal representations: The missing link between faces, text, planktons, and cat breeds. *arXiv preprint arXiv:1701.07275*, 2017.

[2] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, pages 354–370, 2016.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, 2018.

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NeurIPS*, pages 379–387, 2016.

[5] Mark Dredze, Alex Kulesza, and Koby Crammer. Multi-domain learning by confidence-weighted parameter combination. *Machine Learning*, 79(1-2):123–149, 2010.

[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[7] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(Apr):615–637, 2005.

[8] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[10] Georgios Georgakis, Md Alimoor Reza, Arsalan Mousavian, Phi-Hung Le, and Jana Kosecka. Multiview rgb-d dataset for object instance detection. *arXiv preprint arXiv:1609.07826*, 2016.

[11] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 7, 2017.

[16] Peiyun Hu and Deva Ramanan. Finding tiny faces. In *CVPR*, pages 1522–1530, 2017.

[17] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, pages 5001–5009, 2018.

[18] Laurent Itti and Pierre Baldi. A principled approach to detecting surprising events in video. In *CVPR*, volume 1, pages 631–637, 2005.

[19] Wei Jiang, Eric Zavesky, Shih-Fu Chang, and Alex Loui. Cross-domain learning methods for high-level visual concept classification. In *ICIP*, pages 161–164. IEEE, 2008.

[20] Mahesh Joshi, William W Cohen, Mark Dredze, and Carolyn P Rosé. Multi-domain learning: when do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1302–1312. Association for Computational Linguistics, 2012.

[21] Lukasz Kaiser, Aidan N Gomez, Noam Shazeer, Ashish Vaswani, Niki Parmar, Llion Jones, and Jakob Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017.

[22] Tsuyoshi Kato, Hisashi Kashima, Masashi Sugiyama, and Kiyoshi Asai. Multi-task learning via conic programming. In *NeurIPS*, pages 737–744, 2008.

[23] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *ECCV*, pages 158–171. Springer, 2012.

[24] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. *arXiv preprint arXiv:1703.05192*, 2017.

[25] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, page 8, 2017.

[26] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, page 4, 2017.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[28] Anan Liu, Yuting Su, Weizhi Nie, and Mohan S Kankanhalli. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(1):102–114, 2017.

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[30] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[31] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, pages 67–82, 2018.

[32] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016.

[33] Andreas Møgelmose, Mohan M Trivedi, and Thomas B Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Trans. Intelligent Transportation Systems*, 13(4):1484–1497, 2012.

[34] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, and Larry S Davis. Ssh: Single stage headless face detector. In *ICCV*, pages 4885–4894, 2017.

[35] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, June 2016.

[36] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *CVPR*, pages 4293–4302, 2016.

[37] Stephen E Palmer. *Vision science: Photons to phenomenology*. MIT press, 1999.

[38] Vishal M Patel, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015.

[39] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. 2018.

[40] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. In *NeurIPS*, pages 506–516, 2017.

[41] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *CVPR*, pages 8119–8127, 2018.

[42] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1137–1149, 2017.

[46] Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[47] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, page 4, 2017.

[48] Parishwad P Vaidyanathan. *Multirate systems and filter banks*. Pearson Education India, 1993.

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[50] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, pages 2800–2809, 2015.

[51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[52] Jeremy M Wolfe. Visual attention. In *Seeing*, pages 335–386. Elsevier, 2000.

[53] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proc. CVPR*, 2018.

[54] Ke Yan, Mohammadhadi Bagheri, and Ronald M Summers. 3d context enhanced region-based convolutional neural network for end-to-end lesion detection. In *ICCV*, pages 511–519, 2018.

[55] Ke Yan, Xiaosong Wang, Le Lu, Ling Zhang, Adam Harrison, Mohammadhadi Bagheri, and Ronald M Summers. Deep lesion graphs in the wild: relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database. In *IEEE CVPR*, 2018.

[56] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *CVPR*, pages 2129–2137, 2016.

[57] Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. A faster pytorch implementation of faster r-cnn. *https://github.com/jwyang/faster-rcnn.pytorch*, 2017.

[58] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *CVPR*, pages 5525–5533, 2016.

[59] Yongxin Yang and Timothy M Hospedales. A unified perspective on multi-domain and multi-task learning. *arXiv preprint arXiv:1412.7489*, 2014.

[60] Steven Yantis. Control of visual attention. *attention*, 1(1):223–256, 1998.

[61] Alfred L Yarbus. Eye movements during perception of complex objects. In *Eye movements and vision*, pages 171–211. Springer, 1967.

[62] Amir R Zamir, Alexander Sax, William Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, pages 3712–3722, 2018.

[63] Yu Zhang and Qiang Yang. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2017.