Delving into Salient Object Subitizing and Detection

Shengfeng He[1] Jianbo Jiao[2] Xiaodan Zhang[2,3] Guoqiang Han[1] Rynson W.H. Lau[2]
[1]South China University of Technology, China
[2]City University of Hong Kong, Hong Kong
[3]University of Chinese Academy of Sciences, China

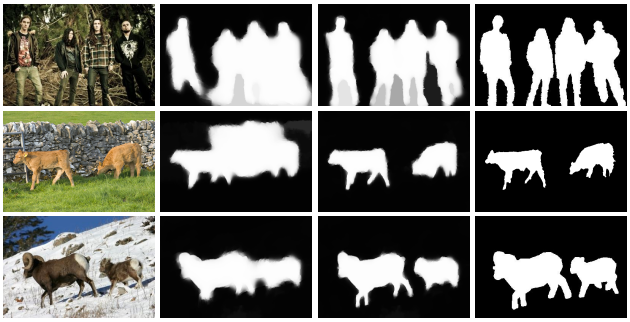
Abstract

Subitizing (i.e., instant judgement on the number) and detection of salient objects are human inborn abilities. These two tasks influence each other in the human visual system. In this paper, we delve into the complementarity of these two tasks. We propose a multi-task deep neural network with weight prediction for salient object detection, where the parameters of an adaptive weight layer are dynamically determined by an auxiliary subitizing network. The numerical representation of salient objects is therefore embedded into the spatial representation. The proposed joint network can be trained end-to-end using back-propagation. Experiments show the proposed multi-task network outperforms existing multi-task architectures, and the auxiliary subitizing network provides strong guidance to salient object detection by reducing false positives and producing coherent saliency maps. Moreover, the proposed method is an unconstrained method able to handle images with/without salient objects. Finally, we show state-of-the-art performance on different salient object datasets.

1. Introduction

The human visual system has an excellent ability in rapidly locating visually distinct objects in a scene, known as visual attention. On the other hand, a human can accurately enumerate up to four objects at a glance without counting. This rapid enumeration of a small number of items is referred to as *subitizing* [13]. These two human inborn abilities can influence each other in an either serial or parallel form within the human visual system [26, 25], and evidences show that numerical and spatial representations are intrinsically interconnected in our brain [11]. These abilities are frequently involved in our daily life for prompt decision making in basic tasks like searching, navigation, and choice making.

However, these biologically-correlated abilities have not been jointly explored in computer vision. Traditional methods focus solely on detecting salient objects. While achieving good results, they suffer from two main problems. First, most methods assume the presence of salient objects, and fail if this is not the case. Second, background distraction is usually the main source of detection errors, leading to excessive detected regions or incorrect connections between objects (see Fig. 1b). These two problems are in fact relevant to the subitizing task. Subitizing knowledge can help predict the existence of dominant objects. Meanwhile, it can also constrain the number of object-like regions to be detected in the saliency map. Although Zhang *et al.* [36, 38] propose to predict the number of salient objects, they only use it to filter images without salient objects. No interactions are considered between the number and the detection of salient objects.



(a) Input (b) w/o subitizing (c) w/ subitizing (d) GT

Figure 1: Our method augments salient object detection with subitizing. Subitizing provides strong guidance to accurately detect salient objects from complex background.

In this paper, we aim to explore the interaction between numerical and spatial representations in the salient object detection task. To this end, we propose a multi-task deep convolution neural network (CNN) with an adaptive weight layer. While this CNN is trained for salient object detection, the weights of its adaptive weight layer are dynamically determined by an auxiliary subitizing network, allowing the layer to encode subitizing knowledge. In this way, a single deep CNN architecture has a dynamical representation space, and it is able to deal with various input contextual information by dynamically predicting weight in the adaptive weight layer. The proposed network is supervised by two different tasks jointly, and it can be trained end-to-end

using back-propagation.

The main contributions of this work are as follows:

- We explore different multi-task networks adapted for integrating multiple sources of knowledge. In particular, we explore how different levels of shared information affect saliency detection performance.
- We design a deep network to detect salient objects with the guidance of subitizing, by introducing an adaptive weight layer. This layer integrates two different tasks by adaptively assigning weights according to the predicted number of salient objects.
- We achieve state-of-the-art performance on all four datasets. Specifically, the proposed method outperforms existing methods on an unconstrained salient object dataset.

To the best of our knowledge, our work is the first to explore the interaction between numerical and spatial representations in a deep model.

2. Related Work

Salient object detection methods can be roughly classified into two categories, hand-crafted and learning based models. As the proposed method belongs to the learning based category, we focus our discussion on relevant deep learning works. A comprehensive literature review can be found in [2].

In recent years, CNNs have been shown to be very effective on various visual recognition tasks, such as image classification [14], semantic segmentation [8] and object detection [9]). We are beginning to see some salient object detection works in these two years. In [10], He *et al.* apply the region-based model to learn the superpixel-wise feature representation, which reduces the computational cost significantly and considers global context information. However, representing a superpixel with the mean color is not informative enough. It is also difficult to fully recover the spatial structure of the image with superpixels.

On the other hand, some methods propose to incorporate both local and global contextual information using CNNs to detect saliency. Wang *et al.* [30] first apply a CNN to extract local patch features to obtain intermediate saliency results, and another CNN to globally integrate the initial saliency map with object proposals. Zhao *et al.* [39] train a two-stream network, one for the local context of the centered region and the other for the global context. A simple fully connected layer is used to combine the two streams. Li *et al.* [17] use a pre-trained CNN as a feature extractor. They concatenate the features obtained from patches of three different scales and feed them to two fully connected layers. These methods, however, apply CNN in a sliding window fashion, resulting in a high computational cost.

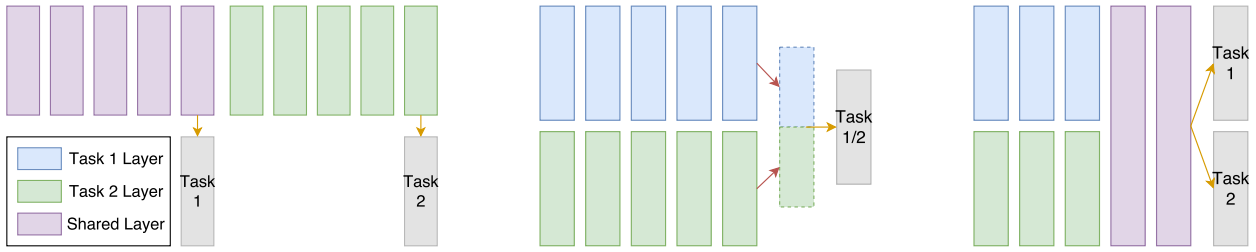
To address the high computational cost, the fully convolutional network (FCN) [21] and deconvolution network [23] are used to generate a saliency map in an end-to-end framework. As they obtain the final result through upsampling from a very coarse prediction, they cannot guarantee accurate segmentation of the saliency map. Different approaches have been proposed to address this problem. Li *et al.* [18] combine the FCN with a segment-wise network using fully connected CRF to obtain a spatial coherent saliency map. Some latest methods [15, 31] aim to refine the resulting saliency map using recurrent networks. Kuen *et al.* [15] use recurrent attentional networks to selectively refine the object boundaries, while Wang *et al.* [31] incorporate recurrent refinement and background priors. However, the added recurrent network may increase the training and testing times. In this paper, we delve into this problem with subitizing guidance and propose three refinement approaches without the recurrent mechanism.

Weight prediction in CNN has been explored in [16] and [24] for zero-shot learning and visual question answering, respectively. Lei *et al.* [16] predict a binary classifier for unseen categories based on the given textual description. Noh *et al.* [24] propose to predict the weights of a fully-connected layer based on the given question in question answering. However, these methods try to predict the weights of the fully-connected layer, which leads to the weight explosion problem. Although a hash trick is applied in [24] to alleviate this problem, performance loss is inevitable. In contrast, we introduce weight prediction in a fully convolutional framework, where the weights can be directly predicted by another network without significantly increasing the number of parameters. We also show that the adaptive weight layer can be trained in a multi-task network. Finally, our work explores the augmentation of one task by another, which leads to better embedding performance.

3. Multi-task Sharing Networks

Given an image, we aim to detect salient objects with the aid of subitizing. An intuitive solution is to jointly train two tasks in a multi-task framework. Here, we explore three multi-task architectures for salient object detection, as shown in Fig. 2.

Cascaded Network: This type of network constructs the layers of two tasks sequentially. The layers of the first task are shared with the second task, while the second task performs prediction with additional layers. This network involves multiple losses. The loss from the second task guides the entire network, while the loss from the first task only supervises the first half of the layers. The cascaded network passes the knowledge in a sequential fashion. However, due to its cascaded architecture, the gradient from the second task may be attenuated in the first half of the layers. In addition, the resolution of the second half of the layers may



(a) Cascaded Network (b) Concatenated Network (c) Shared Network

Figure 2: Multi-task sharing architectures. (a) The cascaded network combines the two tasks in a serial form, where the output features of task 1 are the input of task 2. (b) The concatenated network simply concatenates the output features of two sub-networks for the two tasks. (c) The shared network involves multiple losses during training, and different sharing levels can be achieved by adjusting the number of shared layers.

become too small to retain spatial information, limiting the network to specific tasks.

Concatenated Network: The most straightforward way to integrate multiple sources of knowledge is to concatenate the learned features. The sub-network for each task is typically pre-trained on its own dataset. Once properly trained, these two tasks produce feature maps individually, and concatenated together to form a hyper feature. This hyper feature is then fed to a decision network for the final prediction. The main drawback of this type of network is that both sub-networks are trained individually, and the network can only supervised by one of the tasks. The joint training principle cannot be applied in this type of network.

Shared Network: This architecture might be the most commonly used multi-task architecture, due to its easy-to-adjust characteristics and can be trained jointly with multiple losses. Two tasks are constructed in a parallel form with several shared convolution layers. The number of shared layers may range from one layer (i.e., only the last layer is shared) to all layers (i.e., no parallel layers), depending on applications. This network can be supervised by different forms of losses, to achieve optimal results for both tasks.

Sharing of multi-task knowledge is a common property across these three architectures. However, how much knowledge should be shared remain a question. Using the shared network as an example, there are no principle ways to tell how many shared layers should be set to obtain an optimal multi-task training. To investigate this question, we perform an empirical study using the shared network, by exhaustively tuning the number of shared layers, to find out how shared knowledge affects salient object subitizing and detection performances. We start from sharing no knowledge between two tasks (i.e., two independent networks), which is set as the baseline for the other shared networks. We have a total of five convolution layers, and thus there are five shared network variants. For the subitizing task, the last convolution layer connects with a fully connected layer and then outputs the number of salient objects. For the salient object detection task, the shared network is followed by an upsampling network similar to FCN [21].

Task	Conv5	Conv4-5	Conv3-5	Conv2-5	Conv1-5
Detection (MAE)	-0.53	-0.35	+0.42	+0.31	-0.43
Subitizing (mAP)	+4.17	+5.52	-3.56	-1.32	-0.62

Table 1: Empirical results on the MSO dataset [36], with different numbers of shared layers in the shared network. The values indicate the performance differences between the shared networks and the two independent baselines (i.e., with no shared knowledge). *ConvX-Y* means that the two tasks share parameters between the *X-th* and the *Y-th* convolution layers. As these results do not show a consistent behavior, knowledge sharing may not be the best policy for multi-task learning.

Table 1 shows the empirical results on the MSO dataset [36]. The values indicate the performance differences between the shared networks and the two independent baselines. While sharing the 3rd to 5th convolution layers produces the best performance for salient object detection, sharing the 4th to 5th convolution layers produces the best performance for subitizing. These results reveal a key issue – the performance gain by sharing knowledge is unpredictable, and the influence of different sharing levels depends on the task. A similar observation is also found in the cascaded and concatenated networks. Since enumerating all the possibilities is cumbersome in practice, instead of knowledge sharing, we propose to address the multi-task problem by enriching the representation space of the salient object detection task with dynamic weight assignment.

4. Proposed Network Architecture

This section presents the proposed network architecture and the overall method for salient object detection.

4.1. Overview

The proposed network is a multi-task deep neural network, containing three main components: the salient object detection network, subitizing network, and an adap-

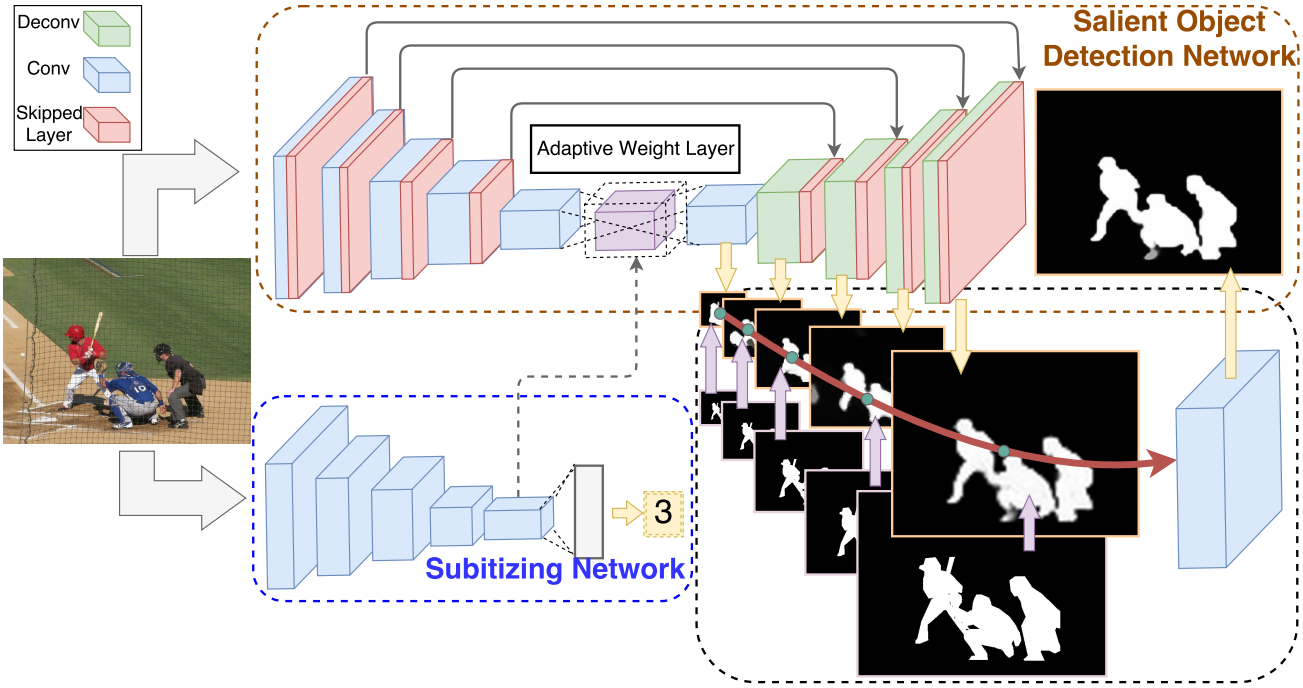


Figure 3: The architecture of the proposed method. The upper part is the salient object detection network, and the bottom part is the subitizing network. An adaptive weight layer is added in the middle of the salient object detection network, where its weights are dynamically determined by the subitizing network, to encode numerical representation into spatial representation. Two refinement approaches are also illustrated: feature-based deconvolution with skipped layers and hierarchical supervision.

tive weight layer. Fig. 3 shows the architecture of the proposed network. The salient object detection network is constructed based on the convolution-deconvolution pipeline. The convolution stage serves as a feature extractor that transforms the input image into a rich feature representation, while the deconvolution stage serves as a shape generator to segment salient objects based on the extracted features. An adaptive weight layer is added between these two stages. It is a convolution layer whose weights are dynamically determined by the auxiliary subitizing network. The subitizing network is trained to predict the number of salient objects and the weights for the adaptive weight layer. The final output is a probability map that indicates how likely each pixel belongs to the salient objects. We discuss these components in more detail in the following subsections.

4.2. Deep Neural Network with Weight Prediction

Given an input image I, the salient object detection network produces a saliency map m from a set of weights θ. The salient object detection is posed as a regression problem, and the saliency value of each pixel (x, y) in m can be described as:

$$m_{x,y} = p(S|R(I, x, y); \theta), \tag{1}$$

where $R(I, x, y)$ corresponds to the receptive field of location (x, y) in m. Once the network is trained, θ is fixed and used to detect salient objects for any input images. However, this set of weights cannot be generalized to all types of input images. This is similar to the denosing problem, where a general denoiser may not perform as well as one that is trained specifically on the noise level of the input images [3]. An intuitive solution to address this generalization problem is to train a task-specific network. However, this requires training numerous networks for different tasks. In addition, this prior information (e.g., noise level) is usually unknown in practice. In contrast, we solve this problem through adaptive weight prediction:

$$m_{x,y} = p(S|R(I, x, y); \theta, \theta_a(n)), \tag{2}$$

where $\theta_a(n)$ is the adaptive weights determined according to the predicted number of salient objects, n, of the input image. In this way, detecting salient objects is not only dependent on the static weights θ, but also the adaptive weights $\theta_a(n)$. In addition, the static weights θ are also trained to interact with the prior information provided by the adaptive weights $\theta_a(n)$. These adaptive weights can be considered as numerical representation of the salient objects.

4.2.1 Predicting Weights with Subitizing

To encode the number of salient objects into the adaptive weights, we implement the numerical feature embedding in a convolution layer. We denote the input feature map of this

convolution layer as f_s. Its corresponding output is then:

$$f_o = W_a(n) f_s + b, \tag{3}$$

where b is the bias, and $W_a(n)$ is the adaptive weight matrix. In this way, the introduced adaptive weight matrix parameterizes this convolution layer as a function of the predicted number of salient objects.

In order to obtain the numerical features of an input image, we apply another network for subitizing. The output features f_n of the subitizing network is directly used as the adaptive weight matrix for salient object detection. The subitizing network can therefore be viewed as a weight prediction network to encode numerical representation into the detection network. Eq. 3 can then be rewritten as:

$$f_o = f_n f_s + b. \tag{4}$$

4.2.2 Back-propagation

The adaptive weight layer can be trained end-to-end using back-propagation. The derivatives of the input and output features in the adaptive weight layer can be computed using standard back-propagation. The derivative of the predicted weights with loss function ℓ is computed as:

$$\frac{\partial \ell}{\partial f_n} = f_s \frac{\partial \ell}{\partial f_o}. \tag{5}$$

4.3. Salient Object Detection Network

As mentioned earlier, we design the salient object detection network based on the convolution-deconvolution pipeline, and these two stages are connected by the adaptive weight layer. The convolution stage is based on the VGG-16 net [27] (with the last classification layer removed), while the deconvolution stage has a mirrored architecture of the convolution stage. This network is first pre-trained for semantic segmentation on the Pascal 2012 dataset [7] without the adaptive weight layer. This convolution-deconvolution pipeline, however, suffers from coarse prediction. We leverage two refinement approaches, skipped features [6] and hierarchical supervision [33], to obtain pixel-level accurate saliency map. The hierarchical supervision guides all the deconvolution layers with the ground truth, and these layers produce the side-output saliency maps. These side-outputs are then fused using a convolution layer to obtain the final result. We further adopt the post-processing techniques in [28] and [31] to obtain compact and boundary-preserved object regions.

4.4. Salient Object Subitizing Network

A human is only able to identify up to 4 salient objects at a glance, effortlessly and consistently. In our work, salient object subitizing is treated as a classification task, and there are five categories in total, representing 0, 1, 2, 3, and 4+ salient objects existed in the image. Similar to the salient object detection network, the subitizing network is based on the VGG-16 net [27]. We modify the size of the input image and the number of filters of the last convolution layer, so that the output feature map can adapt to the size of the weight matrix in the adaptive weight layer. Specifically, the depths of the input and output features of the adaptive weight layer are both 512, and the kernel size is 1×1. As a result, the size of the weight matrix is $1 \times 1 \times 512 \times 512$. Accordingly, we modify the input resolution of the first convolution layer in the subitizing network to 256×256, so that the output resolution of the final convolution layer can be 16×16. The number of filters is set to 1024 in the final convolution layer. Thus, the output feature has a size of $16 \times 16 \times 1024$. This feature is then reshaped to have the same size as the weight matrix.

In addition to connecting to the adaptive weight layer, the last convolution layer is also connected to a fully connected layer followed by a classification layer to predict the existence and the number of salient objects. In other words, the subitizing network is trained by multi-task losses. Before integrating to the salient object detection network, the subitizing network is first pre-trained solely for the subitizing task on the SOS dataset [36].

4.5. Training the Network

The proposed network is trained end-to-end using back-propagation. However, multiple losses are involved in the network, making the training process non-trivial. For the hierarchical supervision, lower layers get mixed gradients if we activate all the hierarchical losses. We leverage a loss weight schedule to overcome this problem. The network is first trained with a loss weight of 1 for the lowest deconvolution layer, and 0 for all the others. During training, the weights of other layers are progressively increased while the lower supervisions are gradually deactivated. In this way, the network begins with learning the coarse representation, leading to a coarse-to-fine learning process.

The two tasks of the subitizing network (i.e., predicting the weights of a layer and predicting the number of salient objects) are inherently different. As such, the pre-trained subitizing knowledge may not be perfectly suitable for the adaptive weight layer. During fine-tuning, we have found that biasing towards the loss from the adaptive weight layer (loss weights of 0.7 vs. 0.3) shows better adaptation in the weight prediction module.

Due to the significant distribution change in the adaptive weight layer with respect to different input images, it is not easy to train an optimized model. Since batch normalization [12] helps generalize the distributions between layers, we apply it to the adaptive weight layer to alleviate this problem. This strategy is also employed in every

convolution and deconvolution layers to prevent the training process from reaching a poor local minima.

5. Experiments

The proposed method is implemented using MatConvnet [29] and tested on a PC with an i7 3.4GHz CPU, an Nvidia Titan X Pascal GPU, and 32GB RAM. The proposed network takes about 0.1s and the post-processing step takes about 2s to process an image of 500×400. We train our model on the training set of the SOS dataset [36], which is the only dataset that contains both the number of salient objects and their bounding box labels. It contains 6,900 images selected from four datasets: MS COCO [20], Pascal VOC 2007 [7], ImageNet [5], and SUN [32]. To accurately learn the object boundaries, we segment all the salient objects from the bounding boxes using the available ground truth segmentations from MS COCO [20] and Pascal VOC 2007 [7]. For those images that are not selected from these two datasets, we keep the original bounding boxes for training. Some data augmentation approaches like cropping, shifting, and flipping are used during training. Our model takes 4 - 5 days for the training to converge.

5.1. Datasets and Evaluation Metrics

As the proposed method is an unconstrained method, it is able to handle images without dominant objects. We evaluate the proposed method on the MSO dataset [36], which is the test set of the SOS dataset [36] for salient object detection. When comparing with the state-of-the-art methods, we further evaluate the proposed method on DUT-OMRON [35], Pascal-S [19], and SOD [22]. The MSO [36] and DUT-OMRON [35] datasets provide only the bounding box ground truth, while the others contain labelled segmentations. Although MSRA10K [4] is the largest dataset for salient object detection, most of the deep learning based models are trained on this dataset or its subset. Therefore, this dataset is not suitable for evaluation.

We compare the proposed method to nine state-of-the-art methods: LEGS [30], MDF [17], DCL [18], RFCN [31], MAP [38], HS [34], GMR [35], MB+ [37], and MST [28]. The first five are deep learning based methods, and the others used hand-crafted features. Among them, only MAP [38] is an unconstrained method. As MAP outputs salient object bounding boxes, we binarize the outputs into saliency maps for evaluation.

Three metrics are used to measure the detection performance: precision-recall (PR), F-measure, and mean absolute error (MAE). The PR curve is computed by thresholding the predicted saliency map into a set of binary masks, and these masks are compared against the ground truth. The F-measure is defined as:

$$F_\beta = \frac{(1+\beta^2) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}, \qquad (6)$$

Metric	**Ours**	Independent	Cascaded	Concatenated	Shared
F-Measure	**0.794**	0.721	0.753	0.767	0.770
MAE	**0.114**	0.143	0.132	0.125	0.124

Table 2: Comparison with different baselines on the Pascal-S dataset [19]. The proposed method outperforms the independent network (no sharing) and the three multi-task sharing networks.

Method	Full		Backgd. only	Sal. Object only	
	Adap. Prec.	MAE	MAE	Adap. Prec.	MAE
Ours	**0.654**	**0.187**	**0.061**	**0.846**	**0.235**
MAP [38]	0.370	0.119	0.068	0.511	0.139

Table 3: Unconstrained salient object detection evaluation on the MSO dataset [36]. The proposed method outperforms MAP [38] not only on images with salient objects, but also on background images.

where β^2 is set to 0.3 to emphasize on precision [1]. An image dependent adaptive threshold [1] is used to compute F-measure, and is set to twice the mean value of the saliency map. The precision with adaptive threshold is also used in the evaluation on the unconstrained dataset (Section 5.2). MAE measures the average pixel-wise error, reflecting the negative saliency assignments.

5.2. Comparison with Baselines

Before comparing with the state-of-the-art methods, we first compare the proposed method with different baseline networks. Four baselines are used, one of them is the independent network (no sharing), the others are the cascaded, concatenated, and shared networks. In the experiments, all these baseline networks compared are trained with the same amount of data (i.e., same number of epoches), using the same refinement and post-processing approaches. All the networks are trained on the SOS dataset [36], and tested on the Pascal-S dataset [19].

Evaluation on Subitizing Embedding. The first evaluation is to examine the effectiveness of subitizing embedding. Table 2 shows the comparison. We can see that integrating subitizing is effective in salient object detection, contributing to an overall F-measure improvement of about 10% and producing 20% less error. This implies that the numerical representation provides strong guidance in spatial representation, helping rectify the final prediction.

Comparison with Multi-task Sharing. Table 2 also shows the comparison with the multi-task sharing networks. We can see that all the networks are effective in improving the detection performance, which shows that subitizing information is useful to the detection task. The cascaded network is not as prominent as the other multi-task sharing networks, as it is too deep to be properly trained. We add multiple losses to the concatenated network, and it can be considered as the closest architecture to ours. This setting

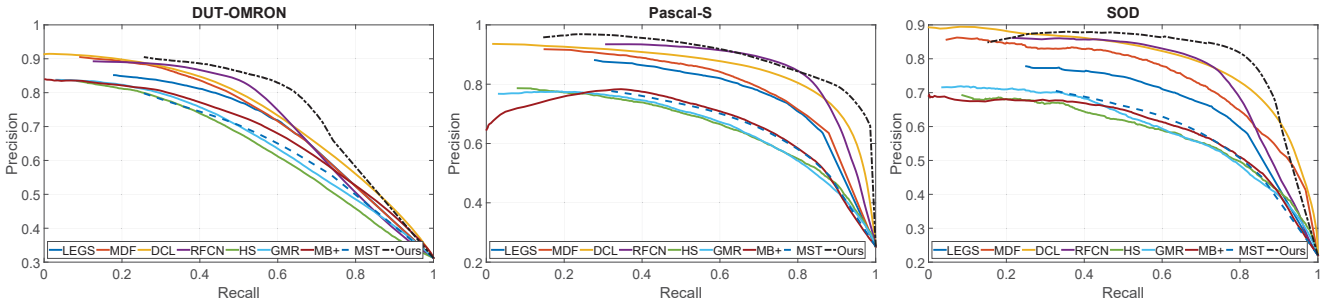


Figure 4: Comparison of precision-recall curves on three datasets. The proposed method consistently outperforms existing methods across all the datasets.

Dataset	Metric	LEGS [30]	MDF [17]	DCL [18]	RFCN [31]	HS [34]	GMR [35]	MB+ [37]	MST [28]	Ours
DUT-OMRON	F-measure	0.608	0.612	0.577	**0.661**	0.499	0.552	0.583	0.569	**0.665**
	MAE	0.242	0.224	0.282	**0.213**	0.305	0.284	0.267	0.246	**0.198**
Pascal-S	F-measure	0.704	0.70	0.718	**0.778**	0.534	0.591	0.614	0.624	**0.794**
	MAE	0.159	0.149	0.127	**0.121**	0.265	0.234	0.230	0.189	**0.114**
SOD	F-measure	0.627	0.656	0.613	**0.718**	0.469	0.538	0.543	0.565	**0.688**
	MAE	0.172	0.152	0.223	**0.130**	0.2667	0.233	0.239	0.187	**0.123**

Table 4: Comparison of F-measure (larger is better) and MAE (smaller is better) on three datasets. The best results are in red, while the second best are in blue.

is better than the cascaded network. The shared network achieves the best result among the multi-task sharing networks, due to its multi-loss supervision and also because we have selected the best architecture from Table 1. The proposed method outperforms all these knowledge sharing networks. The main reason is that the proposed adaptive weight layer dynamically enriches the representation space of the salient object detection network, parameters will be generated according to the input context.

5.3. Comparison with State-of-the-art Methods

Evaluation on the Unconstrained Dataset. As an unconstrained method, we first compare the proposed method with MAP [38] on the MSO dataset [36]. Beside the full MSO dataset, we further report the performance on background images only and on images with salient objects, to better verify the performance of subitizing embedding. As background images do not contain any positive samples (i.e., precision is always zero), F-measure is not valid and MAE is the best metric to measure the detection error. Table 3 shows the adaptive precision and MAE. We can see that even though MSO is more suitable for MAP (as the output of MAP and the ground truth are both bounding boxes), the proposed method achieves much better performance in terms of precision. (Our MAE is higher only because of the difference between our pixel-level segmentation and bounding box ground truth). Specifically, the proposed method performs better on background image identification. The main reason is that MAP does not involve the interaction between the subitizing and detection tasks.

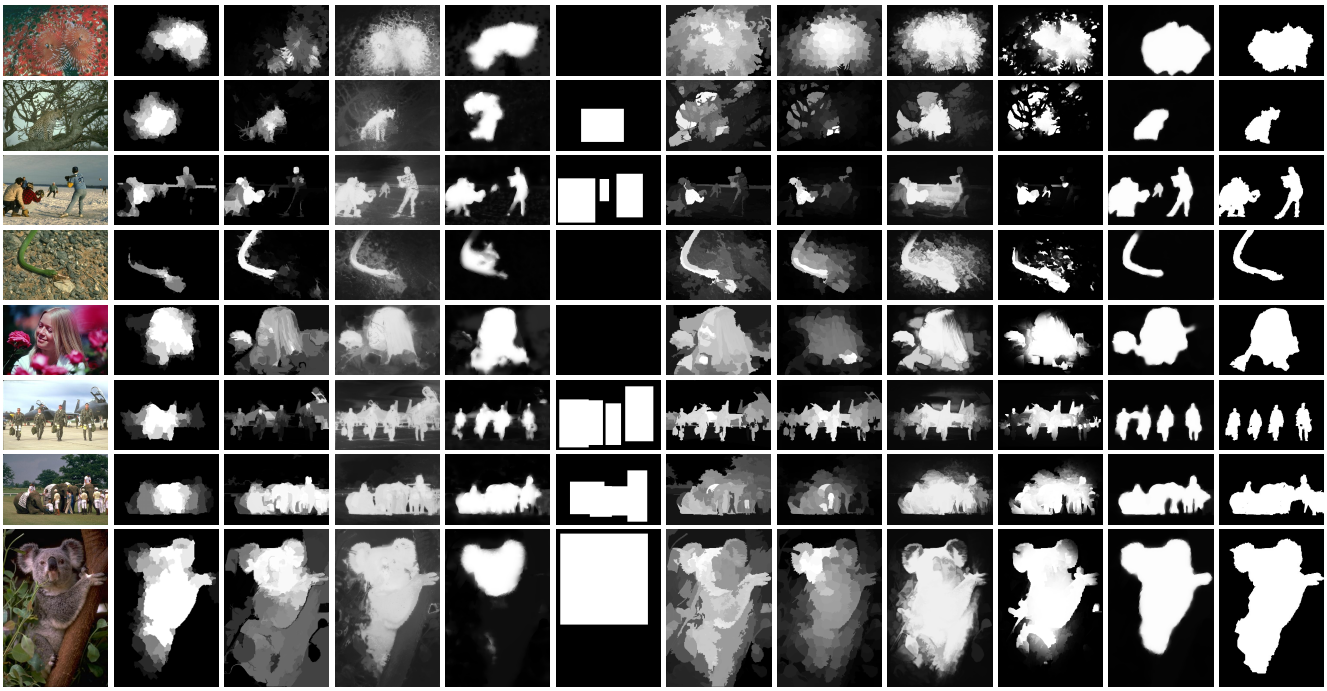
Evaluation on the Traditional Datasets. We then compare with the other state-of-the-art methods on three general datasets. Fig. 4 and Table 4 show the quantitative comparison, and Fig. 5 shows the qualitative comparison. We can see that the proposed method consistently outperforms existing methods on all datasets on the PR curves, F-measure, and MAE. Compared to the other deep learning based methods, we embed subitizing knowledge in salient object detection and thus achieve better performance. As shown in Fig. 5, the proposed method consistently produces compact saliency maps that are closest to the ground truth. Note that in Fig. 5, the unconstrained method MAP [38] may misdetect the non-existence of salient objects.

Method	0	1	2	3	4+
Ours	**96.3%**	**98.0%**	**96.5%**	**97.3%**	**84.1%**
MSO [36]	89.1%	93.3%	92.6%	93.4%	79.0%
Fine-tuned VGG	85.4%	90.5%	89.2%	90.1%	75.9%

Table 5: Subitizing task comparison. The proposed method augmented subitizing strategy achieves the best results on the scenarios with different numbers of salient objects.

5.4. Application: Detection Augmenting Subitizing

As mentioned earlier salient object detection and subitizing are mutually involved in the human visual system. Hence, we are interested to find out if salient object detection may provide effective guidance to subitizing. To this end, we simply swap the objectives of the two networks. While salient object detection is used for weight prediction, subitizing produces our final prediction. Note that we do not need to change the network architecture. We just swap the adaptive weight layer in the detection network with the



(a) Input (b) LEGS (c) MDF (d) DCL (e) RFCN (f) MAP (g) HS (h) GMR (i) MB+ (j) MST (k) Ours (l) GT

Figure 5: Qualitative comparison of the state-of-the-art methods. (b) - (f) are deep learning based methods, while the others use hand-crafted features. The proposed method produces saliency maps closest to the ground truth.

(a) Input (b) GT (c) Ours

Figure 6: Failure cases. The subitizing guidance may sometimes disagree with the ground truth saliency maps on the number of salient objects.

convolution layer in the subitizing network. Table 5 shows the average precision scores. We can see that the proposed method outperforms MSO and a fine-tuned VGG network on the scenarios with different numbers of salient objects. This shows that while subitizing can help improve the performance of saliency object detection, saliency object detection can also help improve the performance of subitizing.

5.5. Failure Cases

Although detecting salient objects with subitizing guidance achieves good performances, the subitizing prediction may not necessarily agree with the ground truth on the number of salient objects. Fig. 6 shows two failure examples. As the subitizing network produces different numbers of salient objects from those of the ground truth, the salient object network outputs different saliency maps.

6. Conclusion

In this paper, we have explored the interactions between numerical and spatial representations in salient object detection. In particular, we delve into the problem of multi-task sharing networks, revealing that their performances are unpredictable and require enumerating all possible architectures to obtain the best one. To address the multi-task problem from a different point of view, we propose a multi-task deep neural network to detect salient objects with the augmentation of subitizing using dynamic weight prediction. Extensive experiments demonstrate that subitizing knowledge provides strong guidance to salient object detection, and the proposed method achieves state-of-the-art performance on four datasets.

Acknowledgements. We would like to thank NVIDIA for generous donation of Titan X Pascal GPU cards for our experiments.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009. 6

[2] A. Borji, M. Cheng, H. Jiang, and J. Li. Salient object detection: A benchmark. *IEEE TIP*, 24(12):5706–5722, 2015. 2

[3] H. Burger, C. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *CVPR*, pages 2392–2399, 2012. 4

[4] M. Cheng, N. Mitra, X. Huang, P. Torr, and S. Hu. Global contrast based salient region detection. *IEEE TPAMI*, 37(3):569–582, 2015. 6

[5] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[6] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015. 5

[7] M. Everingham, S. Eslami, L. van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, Jan. 2015. 5, 6

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE TPAMI*, 35(8):1915–1929, 2013. 2

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[10] S. He, R. Lau, W. Liu, Z. Huang, and Q. Yang. Supercnn: A superpixelwise convolutional neural network for salient object detection. *IJCV*, 115(3):330–344, 2015. 2

[11] E. Hubbard, M. Piazza, P. Pinel, and S. Dehaene. Interactions between number and space in parietal cortex. *Nature Reviews Neuroscience*, 6(6):435–448, 2005. 1

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 5

[13] E. Kaufman, M. Lord, T. Reese, and J. Volkmann. The discrimination of visual number. *The American Journal of Psychology*, 62(4):498–525, 1949. 1

[14] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 2

[15] J. Kuen, Z. Wang, and G. Wang. Recurrent attentional networks for saliency detection. In *CVPR*, June 2016. 2

[16] J. Lei Ba, K. Swersky, S. Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *CVPR*, pages 4247–4255, 2015. 2

[17] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *CVPR*, pages 5455–5463, 2015. 2, 6, 7

[18] G. Li and Y. Yu. Deep contrast learning for salient object detection. In *CVPR*, June 2016. 2, 6, 7

[19] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014. 6

[20] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*. 2014. 6

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 2, 3

[22] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, 2001. 6

[23] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2

[24] H. Noh, P. Hongsuck Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *CVPR*, June 2016. 2

[25] C. Olivers and D. Watson. Subitizing requires attention. *Visual Cognition*, 16(4):439–462, 2008. 1

[26] H. Railo, M. Koivisto, A. Revonsuo, and M. Hannula. The role of attention in subitizing. *Cognition*, 107(1):82–104, 2008. 1

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. 5

[28] W. Tu, S. He, Q. Yang, and S. Chien. Real-time salient object detection with a minimum spanning tree. In *CVPR*, 2016. 5, 6, 7

[29] A. Vedaldi and K. Lenc. MatConvNet – convolutional neural networks for MATLAB. In *ACM Multimedia*, 2015. 6

[30] L. Wang, H. Lu, X. Ruan, and M.-H. Yang. Deep networks for saliency detection via local estimation and global search. In *CVPR*, pages 3183–3192, 2015. 2, 6, 7

[31] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016. 2, 5, 6, 7

[32] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010. 6

[33] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015. 5

[34] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierachical saliency detection. In *CVPR*, 2013. 6, 7

[35] C. Yang, L. Zhang, H. Lu, X. Ruan, and M. Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, 2013. 6, 7

[36] J. Zhang, S. Ma, M. Sameki, S. Sclaroff, M. Betke, Z. Lin, X. Shen, B. Price, and R. Mech. Salient object subitizing. In *CVPR*, pages 4045–4054, 2015. 1, 3, 5, 6, 7

[37] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 fps. In *ICCV*, pages 1404–1412, 2015. 6, 7

[38] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Unconstrained salient object detection via proposal subset optimization. In *CVPR*, 2016. 1, 6, 7

[39] R. Zhao, W. Ouyang, H. Li, and X. Wang. Saliency detection by multi-context deep learning. In *CVPR*, 2015. 2