Salient Object Detection with Dynamic Convolutions

Rohit Venkata Sai Dulam Chandra Kambhamettu
Video/Image Modeling and Synthesis (VIMS) Lab
University of Delaware
{rdulam, chandrak}@udel.edu

Abstract

Convolutional Neural Networks (CNNs) rely on content-independent convolution operations that extract features shared across the entire dataset, limiting their adaptability to individual inputs. In contrast, input-dependent architectures like Vision Transformers (ViTs) can adapt to the specific characteristics of each input. To enhance input adaptability in CNNs, we propose SODDCNet, an encoder-decoder architecture for Salient Object Detection (SOD) that employs large convolutions with dynamically generated weights via the self-attention mechanism. Additionally, unlike other CNN architectures, we utilize multiple large kernels in parallel to segment salient objects of various sizes. To pre-train the proposed model, we combine the COCO and OpenImages semantic segmentation datasets to create a 3.18M image dataset for SOD. Comprehensive quantitative experiments conducted on benchmark datasets demonstrate that SODDCNet performs competitively compared to state-of-the-art methods in SOD and Video SOD. The code and pre-computed saliency maps are provided here.

1. Introduction

In visual data, salient objects are defined as elements that capture immediate attention from observers. The process of identifying these prominent features is referred to as Salient Object Detection (SOD) [2]. Interest in Salient Object Detection (SOD) is rapidly growing because it effectively isolates the most visually distinct objects, providing a reliable set of key points or landmarks. One notable area of research is underwater salient object detection, which is increasingly vital for autonomous underwater robots [21]. These robots must make critical navigation and manipulation decisions based on the relative importance of various objects in their field of view. SOD has also gained significant importance in the field of robotics [38, 41, 45, 61]. For instance, egocentric SOD [61] helps in identifying and segmenting salient objects from a first-person perspective, similar to how an autonomous car or robot operates. Furthermore, saliency-guided localization has become crucial, as shown in works like [41] and [38]. [41] introduces a novel dataset for SOD in traffic scenes, which is essential for autonomous vehicles. [38] presents a SOD-based localization method that allows delivery robots to navigate urban environments, such as campuses and towns, with many unique characteristics. Overall, SOD plays a vital role in localization by enhancing feature extraction and tracking of important objects.

In recent years, salient object detection (SOD) solutions have mainly relied on Convolutional Neural Networks (CNNs) [19] or Vision Transformers (ViTs) [9]. ViT-based models [33, 37] have lately outperformed CNN-based networks [27, 49] due to their adaptability to individual inputs and improved performance with larger datasets. ViTs utilize attention layers that offer a global receptive field, enabling each input element to interact with all others, and input dependence, allowing the model to learn specific features dynamically. In contrast, CNNs use fixed convolutional weights across images, limiting their adaptability. Moreover, although CNNs can theoretically support a large receptive field, their effective receptive field is often small [8]. Thus, we aim to address these limitations in CNNs for SOD.

The resurgence of large-kernel CNNs [8, 34, 36, 53, 59] has led to significant improvements in the ability of CNNs to capture global context in vision tasks. RepLKNet [8] scaled convolutional kernels up to 31×31 using a reparameterization technique that maintains computational efficiency during inference. SLaKNet [34] introduced sparse large-kernel networks with sub-linear scaling, enabling kernels as large as 51×51 without high computational costs. [59] highlights the importance of large kernels that enhance weakly supervised segmentation, where broader receptive fields allow for more accurate localization and feature extraction from noisy or sparsely labeled data. Together, these works emphasize the renewed interest in large-kernel CNNs.

Two main approaches exist to incorporating content-adaptability into CNNS. The first approach uses pooling-based attention mechanisms to adaptively recalibrate feature representations based on the input. Examples include

Model	Params. (M)	Backbone	Pre-training dataset	Pre-training epochs
PiCANet-R [32]	47.22	ResNet-50 [19]	ImageNet [6] (1.28M)	90
BASNet [40]	87.06	ResNet-34 [19]	ImageNet [6] (1.28M)	90
F3-Net [50]	26.50	ResNet-50 [19]	ImageNet [6] (1.28M)	90
LDF [51]	25.15	ResNet-50 [19]	ImageNet [6] (1.28M)	90
VST [33]	44.48	T2T-ViT [60]	ImageNet [6] (1.28M)	310
PSG [58]	25.55	ResNet-50 [19]	ImageNet [6] (1.28M)	90
RCSB [23]	27.90	ResNet-50 [19]	ImageNet [6] (1.28M)	90
CSF-R2Net [17]	36.53	Res2Net [16]	ImageNet [6] (1.28M)	100
EDNet [54]	42.85	ResNet-50 [19]	ImageNet [6] (1.28M)	90
MENet [49]	-	ResNet-50 [19]	ImageNet [6] (1.28M)	90
PGNet [55]	72.70	ResNet-18 [19] & Swin-B [35]	ImageNet22k (14.2M)	90
EnergyT [62]	118.96	Swin [35]	ImageNet [6] (1.28M)	350
TE7[27]	66.27	EfficientNet [46]	ImageNet [6] (1.28M)	350
RMFormer [7]	87.52	Swin-B [35]	ImageNet22k (14.2M)	90
VSCode [37]	74.72	T2T-ViT [60]	ImageNet [6] (1.28M)	310
MDSAM [15]	100.21	SAM [25] (MAE pre-trained ViT-H)	SSL (1.28M) + SA-1B (11M)	800 + 2
SODAWideNet++ [11]	26.58	-	COCO [31] (0.35M)	21
Ours (XL)	78.30	-	Open Images [26] + COCO **(3.18M)**	20
Ours (L)	61.50	-	Open Images [26] + COCO **(3.18M)**	20

Table 1. Above, we list the backbone models and the pre-training pipelines used to pre-train these backbones, which are further used by the current state-of-the-art SOD models.

Squeeze-and-Excitation Networks (SE-Nets) [22] and Convolutional Block Attention Modules (CBAM) [52], which combine channel and spatial attention obtained through spatial or channel pooling to induce input dependence. The second approach alters the convolutional weights using input features. Dynamic Convolutional Neural Networks (Dynamic CNNs) [4], Dynamic Convolutions [64], and CondConv [56] generate coefficients conditioned on the input to adjust convolution weights for each sample. Involution [28] generates a spatial grid of weights for each input location that acts like a feature extractor. These methods offer effective strategies for introducing input adaptability into CNNs.

Most State-of-the-Art (SOD) models rely on pre-trained backbones with extensive training processes, as shown in Table 1. For example, the recent SOD model MDSAM [15] employs a Masked Autoencoding [53] pre-trained Vision Transformer (ViT-H) [9], trained for 800 epochs on ImageNet [6]. Similarly, RMFormer [7] and VSCode [37] use pre-trained backbones with substantial training schedules. However, the disconnect between the pre-training task (image classification) and downstream tasks like SOD might be sub-optimal. [20] demonstrated that models trained from scratch can achieve ImageNet pre-trained performance levels, although requiring longer training. Thus, SODAWideNet++ [11] offers an alternative by modifying the COCO semantic segmentation dataset [31] for SOD pre-training. The resulting dataset and pre-training pipeline were significantly shorter than previous works and led to a competitive model.

The primary contribution of this paper is a novel convolutional neural network (CNN) called ***SODDCNet***, designed for SOD with input-conditioned convolutions. Drawing inspiration from the advantages of large-kernel convolutions, we introduce Dynamic Long-range Units (DLRUs), which consist of multiple stacked convolutions that progressively increase the receptive field. We use convolutions with varying receptive fields at each stage to extract semantic features from diverse contexts. Additionally, to introduce input dependency, we modify the convolutional weights through input conditioning. Unlike previous works [4, 56, 64] that generate a single coefficient for the entire $k \times k$ kernel, we create individual coefficients for each weight within the kernel. This advancement allows for locality-specific feature aggregation. We employ Self-Attention instead of traditional pooling methods to further enhance the model's capability to develop these location-specific weights. Furthermore, we improve our model's performance using a task-specific pre-training dataset. We modified the Open Images semantic segmentation dataset [1, 26] and combined it with the altered COCO dataset [11, 31] to create a new Salient Object Detection dataset consisting of 3.18 million images, which we use for pre-training our model. We summarize our contributions below -

- We propose SODDCNet, a novel CNN model for SOD with large-kernel input-dependent convolutions.
- We propose Dynamic Long Range Units (DLRUs) to extract spatially adaptive convolutional features from various receptive fields.
- We propose a self-attention based weight generation method to induce input dependency.

- We combine the Open Images dataset [1, 26] and modified COCO dataset [11] to create the largest SOD dataset of 3.18M images to pre-train the proposed model.

2. Related Works

2.1. Prior works for Salient Object Detection

PiCANet-R [32] created a contextual attention module that attends to important context locations for each pixel. BASNet [40] utilizes an encoder-decoder network and a boundary refinement network to produce precise saliency predictions with clear boundaries. F3-Net [50] uses a cascaded feedback decoder (CFD) and a cross-feature module to refine semantic features and generate saliency outputs. VST [33] uses a vision transformer-based SOD model as the backbone for Salient Object Detection. PSG [58] uses a loss function to generate auxiliary saliency maps that are used to create accurate saliency maps incrementally. RCSB [23] uses stage-wise feature extraction and novel loss functions to generate saliency predictions. CS-Net [17] uses a flexible convolution module that uses multi-scale features to generate saliency predictions. EDNet [54] presents a unique method of downsampling to obtain a global receptive field that generates high-level features for SOD. LDF [51] proposes a framework that breaks down the original saliency map into body and detail maps for better saliency detection. EnergyT [62] uses an energy-based prior for salient object detection. PGN [55] uses a combination of Resnet [19] and Swin [35] models to generate saliency maps.TR [27] uses an EfficientNet [46] backbone and attention-guided tracing modules to detect salient objects. VSC [37] proposes a foundational model for SOD that uses programmable prompts to generate saliency predictions. MDSAM [15] adapts the SAM model [25] for SOD using feature adaptors. Unlike these works, we propose a deep learning model built explicitly for SOD that is pre-trained on the OpenImages [1, 26] and COCO [11, 31] datasets.

2.2. Prior Works for Video Salient Object Detection

Several prior works have significantly advanced research in video salient object detection (VSOD). SSAV [14] introduced the largest VSOD dataset, DAVSOD, along with a saliency-shift-aware ConvLSTM model that adapts to evolving saliency cues. STVS [3] proposed an optical flow–free 3D CNN framework, leveraging temporal cues within a purely spatial branch to emphasize salient features across frames. WSV [65] explored a weakly supervised approach based on ConvLSTMs, reducing reliance on labor-intensive annotations. DCFNet [63] introduced a dynamic context-sensitive filtering module (DCFM) that uses dynamically generated kernels from consecutive frames to handle video variations effectively. Lastly, MMNet [66] employed a space-time memory (STM)–based encoder-decoder structure to model temporal dynamics in VSOD without relying on explicit optical flow, demonstrating versatility in challenging video scenarios.

2.3. Input-Dependent Convolutions

The convolution operation, with its spatial invariance and inductive biases, is a robust operation that has furthered the state of the art in Computer Vision over the last decade. Nonetheless, the convolution operation has one major drawback: the convolutional weights are static and shared across images, leading to sub-optimal feature learning. Multiple works have tried inducing content adaptivity into CNNs. Squeeze and Excitation networks [22] improve adaptability by adopting a channel-wise weighting mechanism based on input features. Deformable convolutions [5] deviate from standard convolutions by using input-specific pixel locations to extract features instead of the standard grid of locations. Furthermore, the involution operation [28] generates per-pixel weights within a window from the input through a series of transformations, thus emphasizing information in that specific region. [4] use multiple convolutions in parallel per layer and use an SE mechanism to generate weights for each convolutional kernel. Although useful, the above operations obtain contextual information from a smaller window or suppress spatial information excessively through global average pooling, limiting their capabilities. Instead, we utilize the attention operation, which offers a global receptive to generate our convolutional weights.

3. Method

3.1. Overall Architecture

SODDCNet is an encoder-decoder-style network consisting of three components. The first is the neck, which extracts essential local information using a series of six 3×3 convolutions and a stride two maxpooling layers after the second and fourth convolutional layers. The second is the encoder, which consists of two stages. The third component is the decoder, exactly the same as a U-Net [42] decoder. Unlike prior works, we do not use an ImageNet pre-trained backbone for feature extraction. Instead, we pre-train the entire network on the Open Images segmentation dataset by converting the segmentation labels to saliency labels. We explain each stage in detail in the following sections.

3.2. Encoder Stage

Our proposed SODDCNet comprises two encoder stages designed to extract long-range convolutional features through multiple DLRUs. Below, we illustrate the series

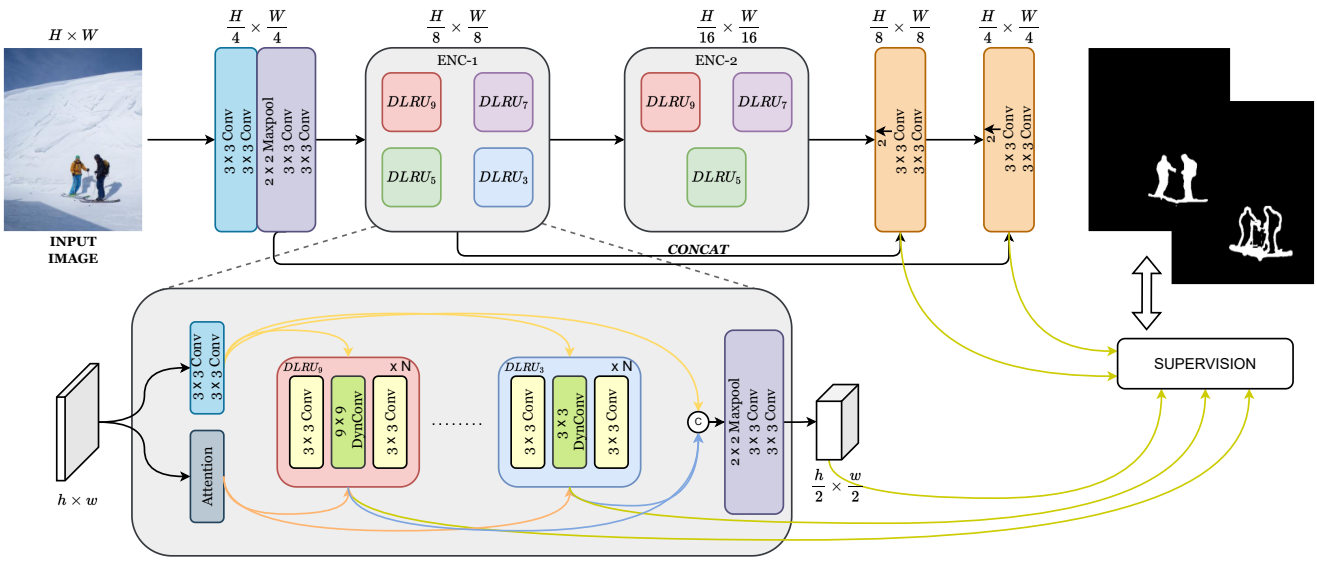

Figure 1. The proposed **SODDCNet** extracts local information through the neck, followed by two encoder stages that capture global and local features through input-conditioned convolutions. Each encoder stage consists of Dynamic Long-Range Units (DLRUs) that consist of dynamic convolutions aided by weights generated by the Attention block. The Attention block uses attention to compute the input-specific weights of different receptive fields. There are two decoder stages where each decoder consists of a bilinear upsampling followed by a concatenation operation and two 3×3 convolutions. [Best viewed in color]

of operations in an encoder stage -

$$
\begin{aligned}
f &= conv(conv(X)) \\
[attn_9, ..., attn_3] &= Attention(X) \\
f_9 ... f_3 &= DLRU_9(X, attn_9) ... DLRU_3(X, attn_3) \\
\bar{f} &= concat[f, f_9, ..., f_3] \\
f_{enc} &= conv(conv(maxpool(\bar{f})))
\end{aligned} \tag{1}
$$

Firstly, the input is spatially transformed through two 3×3 convolutions, each illustrated by $conv$. Each $conv$ also consists of a Batch Normalization operation followed by a ReLU activation function. Simultaneously, we use Attention to generate convolutional weights for all DLRUs in this stage, denoted by $attn_i$. Next, each DLRU takes the transformed features as input and extracts long-range features denoted by f_i, where i denotes the kernel size. Then, features from all the DLRUs are concatenated along with f and sent through a maxpooling layer to reduce spatial resolution, followed by two $conv$ layers for spatial transformation. Finally, we supervise the output from each stage with ground-truth saliency and contour maps.

3.3. Attention Block

Previous works [4, 18] that generate dynamic convolutional kernels use a Global Average Pooling (GAP) layer to condense spatial information and generate convolutional weights. This leads to a significant loss of spatial features, a suboptimal characteristic for a dense prediction task like SOD. Thus, because of its global receptive field, we use Self-Attention (SA) [47] to generate dynamic convolutional weights. The Attention block in our model consists of a series of operations. Firstly, the input's spatial resolution is reduced using Average Pooling, followed by two 3×3 convolutions.

$$
P_{in} = conv(conv(avgPool(X)))
$$

Next, we perform the attention operation on the subsequent feature maps, followed by a convolution operation to change the channel size to equal the window size.

$$
\begin{aligned}
P_{attn} &= SA(P_{in}) \\
attn_{k \times k} &= conv(P_{attn})
\end{aligned}
$$

where SA is the self-attention operation, which is described as -

$$
SA(X) = softmax(\frac{Q \cdot K^T}{\sqrt{d_{dim}}}) \cdot V
$$

K, Q, and V are the key, query, and value tensors generated from the input X. d_{dim} indicates the embedding dimension. Once we compute the attention features, we obtain $attn_{k \times k} \in R^{k \times k \times h' \times w'}$ using a 1×1 convolution. Each $attn_{k \times k}$ feature map is a per-pixel mask generated for each convolution kernel with spatial resolution $k \times k$. h' and w' indicate the height and width of the mask features. We use a different convolution layer to obtain the weights for each k. Figure 1 visually illustrates the entire procedure.

3.4. Dynamic Long Range Unit (DLRU)

Traditional convolutions extract features by scanning the entire feature map through fixed-weight matrices, which search for common patterns across the dataset. Although this leads to spatial invariance, these spatially shared filters struggle to identify sophisticated characteristics in each input. Furthermore, the most famous CNN architectures [19, 44] for SOD rely on tiny 3×3 convolutions, which significantly restrict the receptive field in each layer.

We address these limitations by generating input-specific convolutional weights using self-attention, inducing spatial adaptability in a convolution operation. Additionally, we used multiple large-kernel convolutions in parallel to capture features from different receptive fields at once. As seen in Figure 1, each DLRU contains four components. Two 3×3 convolutions to pre-process and post-process feature maps, a $k \times k$ standard convolution to reduce channel size and spatial resolution, and finally the dynamic convolution dyn_conv.

$$
\begin{aligned}
f_{pre} &= conv(f_k^{i-1}) \\
f_{lower} &= conv_{k\times k}(f_{pre}) \\
f_{dyn} &= dyn\_conv_{k\times k}(f_{lower}, attn_{k\times k}) \\
f_{k\times k} &= act(norm(f_{dyn})) \\
f_k^i &= conv(f_{k\times k})
\end{aligned}
$$

We stack multiple DLRUs on top of each other, which is denoted by $i \in [1, ..., N]$. To reduce computational complexity, the spatial resolution of the input is reduced using a strided convolution with the same kernel size as the dynamic convolution. After the dyn_conv operation, a bilinear upsampling operation increases the spatial resolution to bring it to the input size. We visualize the outputs from each DLRU in both the encoding stages in Figure 5.

3.5. Dynamic Convolution

To address the limitations of the traditional convolution operation, we propose Dynamic Convolution, a context-adaptive operation designed to capture highly localized, spatially varying features. In dynamic convolution, each pixel location in the input feature map receives an individualized kernel, adaptively computed from the local spatial and semantic context, thus enabling per-pixel modulation of feature extraction.

Let $X \in R^{H\times W\times C_i}$ denote the input feature map with a height H, width W, and C_i input channels. A convolutional weight $W \in R^{c_o\times c_i\times k\times k}$ is a set of weight matrices with a kernel size of $k \times k$. Thus, a standard convolution operation for each pixel location P_{ij} in the output Y can be written as -

$$
Y(P_{ij}) = \sum_{P_n \in S} W_{P_n} \cdot X(P_{ij} + P_n) \tag{2}
$$

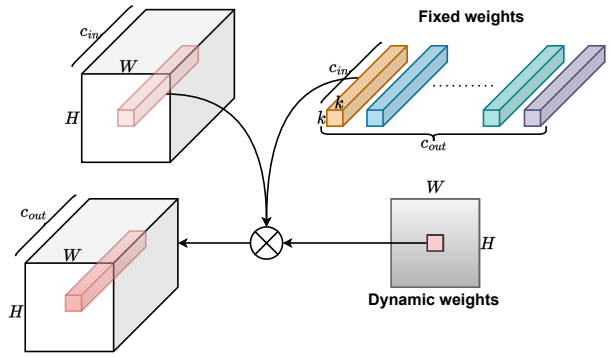

Figure 2. The Dynamic convolution operation consists of three sets of tensors, fixed weights (W), the weights generated by the Attenion block (W_{dyn}), and the input tensor. The Attention block generates per-pixel masks, and each such mask is multiplied by every tensor in W, generating per-pixel convolutional weights. We use these weights to generate long-range dynamic features.

where $S = \{\langle \lfloor -k/2 \rfloor, \lfloor -k/2 \rfloor \rangle, ..., \langle \lfloor k/2 \rfloor, \lfloor k/2 \rfloor \rangle\}$ refers to the offsets in the neighborhood of the window.

Our dynamic convolution implementation modifies 2 by adding a dynamic weight parameter $W^{dyn} \in R^{k\times k\times h\times w}$.

$$
Y(P_{ij}) = \sum_{P_n \in S} W_{P_n} \cdot X(P_{ij} + P_n) \cdot W_{P_n}^{dyn} \tag{3}
$$

Each pixel P_{ij} in the spatial domain is associated with a unique $k\times k$ weighting map from W^{dyn}, as shown in Figure 2. This design allows the network to adaptively adjust convolutional parameters on a per-pixel basis, enabling highly localized feature extraction that can better capture complex spatial dependencies compared to a single, static kernel.

3.6. Loss Function

Feature maps of each DLRU, encoder stage (ENC), and decoder stage (DEC) produce saliency and contour outputs. To supervise these outputs, we follow the same loss function proposed by [11], consisting of binary-cross-entropy loss, dice loss, IoU loss, and L1 loss.

$$
\begin{aligned}
L_{salient} = \sum_{i=1}^{4}(L_{DLRU_{1,i}}^{sal}) + \sum_{i=1}^{3}(L_{DLRU_{2,i}}^{sal}) + \\
\sum_{i=1}^{2}(L_{ENC_{(i)}}^{sal} + L_{DEC_{(i)}}^{sal})
\end{aligned} \tag{4}
$$

We use four DLRUs in the first encoder stage and three DLRUs in the second encoder stage, denoted by the first and second loss terms in Equation 4, respectively. Similarly, the total contour loss is written as:

Methods/Datasets	DUTS-TE [48]				DUT-OMRON [57]				HKU-IS [29]				ECSSD [43]				PASCAL-S [30]			
	F_{max}	MAE	E_m	S_m	F_{max}	MAE	E_m	S_m	F_{max}	MAE	E_m	S_m	F_{max}	MAE	E_m	S_m	F_{max}	MAE	E_m	S_m
PiCANet-R [32]$_{18}$	0.860	0.051	0.862	0.869	0.803	0.065	0.841	0.832	0.918	0.043	0.936	0.904	0.935	0.046	0.913	0.917	0.868	0.078	0.837	0.852
BASNet [40]$_{19}$	0.860	0.048	0.884	0.866	0.805	0.056	0.869	0.836	0.928	0.032	0.946	0.909	0.942	0.037	0.921	0.916	0.860	0.079	0.850	0.834
F3-Net [50]$_{20}$	0.891	0.035	0.902	0.888	0.813	0.053	0.870	0.838	0.937	0.028	0.953	0.917	0.945	0.033	0.927	0.924	0.882	0.064	0.863	0.857
LDF [51]$_{20}$	0.898	0.034	0.910	0.892	0.820	0.051	0.873	0.838	0.939	0.027	0.954	0.919	0.950	0.034	0.925	0.924	0.887	0.062	0.869	0.859
VST [33]$_{21}$	0.890	0.037	0.892	0.896	0.825	0.058	0.861	0.850	0.942	0.029	0.953	0.928	0.951	0.033	0.918	0.932	0.890	0.062	0.846	0.871
PSG [58]$_{21}$	0.886	0.036	0.908	0.883	0.811	0.052	0.870	0.831	0.938	0.027	0.958	0.919	0.949	0.031	0.928	0.925	0.886	0.063	0.863	0.858
RCSB [23]$_{22}$	0.889	0.035	0.903	0.878	0.810	0.045	0.856	0.820	0.938	0.027	0.954	0.918	0.944	0.033	0.923	0.921	0.886	0.061	0.858	0.857
CSF-R2Net [17]$_{20}$	0.890	0.037	0.897	0.890	0.815	0.055	0.861	0.838	0.935	0.030	0.952	0.921	0.950	0.033	0.928	0.930	0.886	0.069	0.855	0.862
EDNet [54]$_{22}$	0.895	0.035	0.908	0.892	0.828	0.048	0.876	0.846	0.941	0.026	0.956	0.924	0.951	0.032	0.929	0.927	0.891	0.065	0.867	0.860
MENet [49]$_{23}$	0.913	0.028	0.921	0.905	0.834	0.045	0.882	0.850	0.948	0.023	0.960	0.927	0.955	0.031	0.925	0.928	0.901	0.057	0.866	0.868
PGNet [55]$_{22}$	0.917	0.027	0.922	0.911	0.835	0.044	0.887	0.855	0.948	0.024	0.961	0.929	0.960	0.027	0.932	0.918	0.904	0.054	**0.878**	0.874
EnergyT [62]$_{21}$	0.910	0.029	0.909	0.909	0.839	0.050	0.886	0.858	0.947	0.023	0.961	0.930	0.959	**0.023**	**0.933**	0.942	0.900	0.055	0.869	**0.876**
TE7 [27]$_{22}$	**0.927**	**0.022**	**0.934**	0.919	0.828	0.048	0.876	0.846	0.951	**0.020**	0.964	0.934	0.959	0.026	0.927	0.936	**0.911**	**0.049**	**0.880**	**0.880**
SODAWideNet++ [11]$_{24}$	0.915	0.030	0.916	0.910	0.847	0.046	0.896	0.868	0.949	0.025	0.960	0.932	0.957	0.030	0.927	0.935	0.900	0.063	0.868	0.874
RMFormer [7]$_{23}$	**0.931**	**0.023**	**0.933**	**0.925**	**0.861**	**0.040**	**0.904**	**0.877**	**0.957**	**0.019**	**0.968**	**0.940**	0.964	**0.021**	**0.934**	0.949	-	-	-	-
VSCode [37]$_{24}$	**0.931**	**0.024**	**0.931**	**0.926**	**0.861**	**0.042**	**0.899**	**0.876**	**0.957**	**0.021**	0.965	**0.940**	**0.965**	**0.021**	**0.934**	**0.949**	**0.912**	**0.051**	0.870	**0.885**
MDSAM [15]$_{24}$	**0.927**	**0.024**	0.929	0.920	**0.868**	**0.039**	**0.908**	**0.878**	**0.956**	**0.020**	**0.967**	**0.941**	**0.968**	**0.021**	**0.937**	**0.948**	0.903	0.055	0.874	**0.880**
SODDCNet-XL (Ours)	**0.928**	**0.024**	0.929	**0.923**	**0.858**	**0.042**	**0.904**	**0.878**	**0.956**	**0.021**	**0.966**	**0.941**	0.964	**0.024**	0.932	**0.945**	**0.912**	**0.051**	0.876	**0.885**
SODDCNet-L (Ours)	0.927	0.025	0.928	0.922	0.852	0.043	0.902	0.872	**0.955**	0.022	0.964	**0.938**	**0.966**	0.025	0.932	0.944	**0.915**	**0.052**	**0.882**	**0.885**

Table 2. Quantitative comparison of our method with 17 other state-of-the-art models in terms of F_{max}, MAE, E_m, and S_m measures across different datasets. Best, second, and third results are highlighted in Red, Blue, and Green, respectively.

Figure 3. Qualitative comparison. Our model distinctly generates masks on five samples compared to the ground truth and state-of-the-art methods. For example, 1) both the monkeys, the foxes, and the birds, 2) the entire man without the chair, and 3) the entire building.

$$L_{contour} = \sum_{i=1}^{4}(L^{con}_{DLRU_{1,i}}) + \sum_{i=1}^{3}(L^{con}_{DLRU_{2,i}}) + \sum_{i=1}^{2}(L^{con}_{ENC_{(i)}} + L^{con}_{DEC_{(i)}}) \quad (5)$$
$$L^{con} = 0.001 \cdot L_{BCE} + L_{dice}$$

L_{BCE} and L_{dice} are the binary cross-entropy and dice loss, respectively. Finally, from Equations 4, and 5, the total loss to train our model is given as

$$L_{total} = L_{salient} + L_{contour} \quad (6)$$

4. Experiments and Results

4.1. SOD Datasets

To pre-train our model, we combine the modified OpenImages dataset [1, 26] of 944K annotated images and the modified COCO dataset from [10], creating a 1.28M dataset. We further augment the Open Images dataset to create a 3.18M dataset, which we use to pre-train our model. Then, we fine-tune our model on the DUTS [48] dataset, which contains

Dataset & Metric		SSAV [14]	WSV [65]	STVS [3]	DCF [63]	MMN [66]	Ours
DAVIS [39]	$S\uparrow$	0.893	0.828	0.892	0.914	0.897	0.914
	$M\downarrow$	0.028	0.037	0.023	0.016	0.020	0.016
	$F\uparrow$	0.861	0.779	0.865	0.900	0.877	0.901
DAVSOD [14]	$S\uparrow$	0.724	0.705	0.744	0.741	0.777	0.781
	$M\downarrow$	0.092	0.103	0.086	0.074	0.065	0.061
	$F\uparrow$	0.603	0.605	0.650	0.660	0.708	0.705
DAVSOD-N [14]	$S\uparrow$	0.661	0.633	0.675	0.686	0.688	0.708
	$M\downarrow$	0.117	0.14	0.108	0.094	0.088	0.094
	$F\uparrow$	0.509	0.485	0.540	0.574	0.555	0.603
DAVSOD-D [14]	$S\uparrow$	0.619	0.572	0.623	0.613	0.622	0.657
	$M\downarrow$	0.114	0.163	0.097	0.090	0.089	0.079
	$F\uparrow$	0.399	0.383	0.409	0.403	0.418	0.482

Table 3. Performance on four video SOD benchmarks (DAVIS [39], DAVSOD [14], DAVSOD-N [14], DAVSOD-D [14]) with state-of-the-art methods. Metrics S (S-measure), M (MAE), F (F-measure). Higher is better for S/F, lower is better for M.

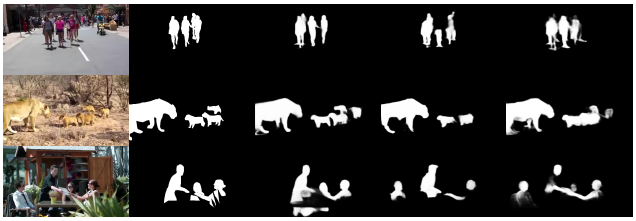

Figure 4. Qualitative comparison. Our model distinctly generates masks on three samples compared to the ground truth and state-of-the-art methods. For example, 1) all the humans and 2) all the animals.

10,553 images for training. We further augment it to obtain a training dataset of 31,659 images. We use five datasets to evaluate the proposed model. They are DUTS-Test [48] consisting of 5019 images, DUT-OMRON [57] which consists of 5168 images, HKU-IS [29] which consists of 4447 images, ECSSD [43] which consists of 1000 images and PASCAL-S [30] dataset consisting of 850 images.

4.2. VSOD Datasets

To train our model for Video Salient Object Detection, we combine the train splits of the DAVIS [39] and DAVSOD [14] datasets. DAVIS consists of 30 videos for training and 20 videos for testing. We specifically utilize the 480p resolution data. DAVSOD consists of a train set and three different test sets: DAVSOD-Easy (DAVSOD) with 35 videos, DAVSOD-Normal (DAVSOD-N) with 25 videos, and DAVSOD-Difficult (DAVSOD-D) with 20 videos.

4.3. Implementation Details

We provide two models *SODDCNet-XL* with 78.3M parameters and *SODDCNet-L* with 61.5M parameters. The smaller model replaces two 3×3 convolutions by a 3×3 convolution followed by a 1×1 convolution. Also, the 3×3 convolutions in each DLRU are replaced by a 1×1 convolution. For OpenImages pre-training, we train our model for 20 epochs. We use a cosine learning rate scheduler with a two-epoch warmup. For SOD fine-tuning, we train our model for a further 11 epochs with a starting LR of 0.001, multiplied by 0.5 after five epochs. Images are resized to 384×384 for training and testing. The first stage uses convolution kernels of sizes $[9, 7, 5, 3]$. The kernel sizes for the second encoder stage are $[9, 7, 5]$. We finalized the kernel sizes in the encoder through experimentation. We use Adam optimizer [24] with its default parameters to update the weights. The evaluation metrics for comparing our works with prior works are the Mean Absolute Error(MAE), maximum F-measure, the E-measure [13], and the S-measure [12]. For VSOD implementation, we use the DUTS-trained model since prior works use DUTS pretraining as a preliminary step before VSOD training. We use the same training regime used in SOD training for VSOD training. We report the maximum F-measure, the S-measure [12], and MAE as evaluation metrics.

4.4. Quantitative Results

Table 2 presents the performance of our SODDCNet model compared to other state-of-the-art SOD models. Despite a significantly smaller pretraining pipeline using OpenImages, our model competes well with other more recent transformer-based models like VSCode, RMFormer, and MDSAM. Remarkably, SODDCNet performs very competitively against the most recent work, MDSAM, despite using an 800-epoch pre-trained backbone. Also, SODDCNet outperforms the state-of-the-art CNN-based SOD model Tracer(TE7) on DUT-OMRON, HKU-IS, and ECSSD datasets. It consistently ranks in the top three across all evaluation metrics and datasets. Similarly, for VSOD, we compare against five state-of-the-art models on four benchmark datasets. SODDCNet beats all the other models by a significant margin on most metrics. Especially on the DAVSOD-N and DAVSOD-D datasets, our model exceeds the performance of previous sota models by at least 2%. These metrics demonstrate the efficacy of our proposed SODDCNet and the utilization of a pre-training pipeline closely related to the target task.

4.5. Qualitative Results

Figure 3 illustrates the visual results of our model against other state-of-the-art SOD models. The salient objects in all the figures seem to blend with their respective backgrounds, making it difficult to identify their boundaries precisely. Nonetheless, our model outperforms all the other models by consistently segmenting the salient objects. For example, most models failed to detect the larger monkey in the first row, whereas our model precisely identified it. Similarly, none of the other models segmented the smaller animal in the second row, whereas our method accurately captured both animals. The same is the case with the third

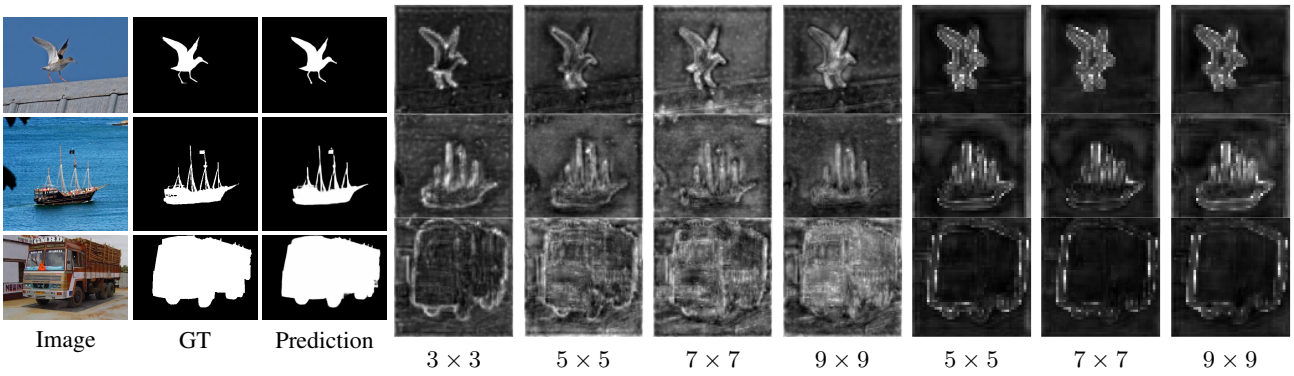

Figure 5. The above visual displays the intermediate features generated by each DLRU with a specific receptive field. In the first encoder stage, the smaller convolution kernels 3×3 and 5×5 identify edges, whereas the larger kernels capture the entire object. However, in the second encoder stage, all the kernels consistently concentrate on the salient object.

row, where most models failed to differentiate the smaller white bird from the background snow. In the fourth row, all the other models could not distinguish between the human and the chair, ending up segmenting both objects, whereas our model precisely segmented only the human and not the chair. A similar observation from the last row shows that our model possesses superior background-foreground differentiation ability than prior works. Similar observations can be made from Figure 4, which contains visual VSOD results.

5. Ablation Studies

In this section, we understand the behavior of various components of the proposed network. All the results are reported on the DUTS test set. We use MAE and F_{max} metrics for evaluating the different configurations.

5.1. Influence of different convolutions kernels

In this section, we consider three different convolutional kernels, traditional large kernels, dilated convolutions, and small kernels, to study their influence on our proposed model as shown in table 4. We start with attention-generated large convolution kernels in the proposed SODDCNet. To understand the importance of these larger kernels, we replace them with 3×3 dilated convolutions with the same receptive field and traditional 3×3 convolutions. These three scenarios are named as S_1, S_2, and S_3. S_1 corresponds to the SODDCNet setting, S_2 replaces all the convolution layers with 3×3 dilated convolution with appropriate dilation rates with the same receptive field as in S_1. The final setting S_3 consists of the same number of traditional 3×3 convolutions.

5.2. Impact of Attention-generated weights

Each Dynamic convolution is a feature extraction unit in a DLRU that contains the attention-generated weights and standard convolutional weights. Using only the static convolution translates our network to a traditional CNN whose weights are content-independent and constant during the inference. We report performance with and without attention-generated convolutional weights in Table 5.

Setting	MAE	F_{max}
Large kernels (S_1)	0.025	0.927
Dilated Convolutions (S_2)	0.030	0.917
Small kernels (S_3)	0.030	0.915

Table 4. Perfomance when using attention-generated large, dilated, and small convolutions.

Setting	MAE	F_{max}
w.o Dynamic Weights	0.028	0.923
w. Dynamic Weights	0.025	0.927

Table 5. Performance of SODDCNet with and without the dynamic weights generated by self-attention.

6. Conclusion

Our proposed method, SODDCNet, integrates large kernel convolutions with attention-based weight generation. We create convolutional weights that extract input-specific semantic features from multiple receptive fields using Dynamic Long Range Units (DLRUs). Specifically, we generate per-pixel masks that guide the convolutional weights, promoting content adaptability. To pre-train our model, we merge the Open Images semantic segmentation dataset with the COCO dataset, resulting in a comprehensive dataset of 3.18 million images. Our model demonstrates competitive performance compared to state-of-the-art models, even those with extensive pre-training schedules, across two tasks and nine benchmark datasets. Additionally, it achieves superior results in qualitative assessments.

References

[1] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019. 2, 3, 6

[2] Ali Borji. What is a salient object? a dataset and a baseline model for salient object detection. *IEEE Transactions on Image Processing*, 24(2):742–756, 2014. 1

[3] Chenglizhao Chen, Guotao Wang, Chong Peng, Yuming Fang, Dingwen Zhang, and Hong Qin. Exploring rich and efficient spatial temporal interactions for real-time video salient object detection. *IEEE Transactions on Image Processing*, 30:3995–4007, 2021. 3, 7

[4] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11030–11039, 2020. 2, 3, 4

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[7] Xinhao Deng, Pingping Zhang, Wei Liu, and Huchuan Lu. Recurrent multi-scale transformer for high-resolution salient object detection. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7413–7423, 2023. 2, 6

[8] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022. 1

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[10] Rohit Venkata Sai Dulam and Chandra Kambhamettu. Sodawidenet-salient object detection with an attention augmented wide encoder decoder network without imagenet pre-training. In *International Symposium on Visual Computing*, pages 93–105. Springer, 2023. 6

[11] Rohit Venkata Sai Dulam and Chandra Kambhamettu. Sodawidenet++: Combining attention and convolutions for salient object detection. *arXiv preprint arXiv:2408.16645*, 2024. 2, 3, 5, 6

[12] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *Proceedings of the IEEE international conference on computer vision*, pages 4548–4557, 2017. 7

[13] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421*, 2018. 7

[14] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8554–8564, 2019. 3, 7

[15] Shixuan Gao, Pingping Zhang, Tianyu Yan, and Huchuan Lu. Multi-scale and detail-enhanced segment anything model for salient object detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 9894–9903, 2024. 2, 3, 6

[16] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE TPAMI*, 2020. 2

[17] Shang-Hua Gao, Yong-Qiang Tan, Ming-Ming Cheng, Chengze Lu, Yunpeng Chen, and Shuicheng Yan. Highly efficient salient object detection with 100k parameters. In *ECCV*, 2020. 2, 3, 6

[18] David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 3, 5

[20] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4918–4927, 2019. 2

[21] Lin Hong, Xin Wang, Gan Zhang, and Ming Zhao. Usod10k: a new benchmark dataset for underwater salient object detection. *IEEE transactions on image processing*, 2023. 1

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 2, 3

[23] Yun Yi Ke and Takahiro Tsubono. Recursive contour-saliency blending network for accurate salient object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2940–2950, 2022. 2, 3, 6

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[25] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2, 3

[26] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020. 2, 3, 6

[27] Min Seok Lee, WooSeok Shin, and Sung Won Han. Tracer: Extreme attention guided salient object tracing network (student abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12993–12994, 2022. 1, 2, 3, 6

[28] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12321–12330, 2021. 2, 3

[29] G. Li and Y. Yu. Visual saliency based on multiscale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5455–5463, 2015. 6, 7

[30] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 280–287, 2014. 6, 7

[31] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2, 3

[32] Nian Liu, Junwei Han, and Ming-Hsuan Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3089–3098, 2018. 2, 3, 6

[33] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4722–4732, 2021. 1, 2, 3, 6

[34] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 1

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 3

[36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 1

[37] Ziyang Luo, Nian Liu, Wangbo Zhao, Xuguang Yang, Dingwen Zhang, Deng-Ping Fan, Fahad Khan, and Junwei Han. Vscode: General visual salient and camouflaged object detection with 2d prompt learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17169–17180, 2024. 1, 2, 3, 6

[38] Jooyong Park, Jungwoo Lee, Euncheol Choi, and Younggun Cho. Salience-guided ground factor for robust localization of delivery robots in complex urban environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1701–1708. IEEE, 2024. 1

[39] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Computer Vision and Pattern Recognition*, 2016. 7

[40] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7479–7489, 2019. 2, 3, 6

[41] Yu Qiu, Yuhang Sun, Jie Mei, Lin Xiao, and Jing Xu. Salient object detection in traffic scene through the tsod10k dataset. *arXiv preprint arXiv:2503.16910*, 2025. 1

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[43] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4): 717–729, 2015. 6, 7

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[45] Kechen Song, Jie Wang, Yanqi Bao, Liming Huang, and Yunhui Yan. A novel visible-depth-thermal image dataset of salient object detection for robotic visual perception. *IEEE/ASME Transactions on Mechatronics*, 28(3):1558–1569, 2022. 1

[46] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2, 3

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[48] Lijun Wang, Huchuan Lu, Yifan Wang, Mengyang Feng, Dong Wang, Baocai Yin, and Xiang Ruan. Learning to detect salient objects with image-level supervision. In *CVPR*, 2017. 6, 7

[49] Yi Wang, Ruili Wang, Xin Fan, Tianzhu Wang, and Xiangjian He. Pixels, regions, and objects: Multiple enhancement for salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10031–10040, 2023. 1, 2, 6

[50] Jun Wei, Shuhui Wang, and Qingming Huang. F^3net: fusion, feedback and focus for salient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12321–12328, 2020. 2, 3, 6

[51] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13025–13034, 2020. 2, 3, 6

[52] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In

Proceedings of the European conference on computer vision (ECCV), pages 3–19, 2018. 2

[53] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16133–16142, 2023. 1, 2

[54] Yu-Huan Wu, Yun Liu, Le Zhang, Ming-Ming Cheng, and Bo Ren. Edn: Salient object detection via extremely-downsampled network. *IEEE Transactions on Image Processing*, 2022. 2, 3, 6

[55] Chenxi Xie, Changqun Xia, Mingcan Ma, Zhirui Zhao, Xiaowu Chen, and Jia Li. Pyramid grafting network for one-stage high resolution saliency detection. In *CVPR*, 2022. 2, 3, 6

[56] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in neural information processing systems*, 32, 2019. 2

[57] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3166–3173, 2013. 6, 7

[58] Sheng Yang, Weisi Lin, Guosheng Lin, Qiuping Jiang, and Zichuan Liu. Progressive self-guided loss for salient object detection. *IEEE Transactions on Image Processing*, 30: 8426–8438, 2021. 2, 3, 6

[59] Shunsuke Yasuki and Masato Taki. Cam back again: Large kernel cnns from a weakly supervised object localization perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 341–351, 2024. 1

[60] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 558–567, 2021. 2

[61] Hao Zhang, Haoran Liang, Xing Zhao, Jian Liu, and Ronghua Liang. Salient object detection in egocentric videos. *IET Image Processing*, 18(8):2028–2037, 2024. 1

[62] Jing Zhang, Jianwen Xie, Nick Barnes, and Ping Li. Learning generative vision transformer with energy-based latent space for saliency prediction. In *2021 Conference on Neural Information Processing Systems*, 2021. 2, 3, 6

[63] Miao Zhang, Jie Liu, Yifei Wang, Yongri Piao, Shunyu Yao, Wei Ji, Jingjing Li, Huchuan Lu, and Zhongxuan Luo. Dynamic context-sensitive filtering network for video salient object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1553–1563, 2021. 3, 7

[64] Yikang Zhang, Jian Zhang, Qiang Wang, and Zhao Zhong. Dynet: Dynamic convolution for accelerating convolutional neural networks. *arXiv preprint arXiv:2004.10694*, 2020. 2

[65] Wangbo Zhao, Jing Zhang, Long Li, Nick Barnes, Nian Liu, and Junwei Han. Weakly supervised video salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16826–16835, 2021. 3, 7

[66] Xing Zhao, Haoran Liang, Peipei Li, Guodao Sun, Dongdong Zhao, Ronghua Liang, and Xiaofei He. Motion-aware memory network for fast video salient object detection. *IEEE Transactions on Image Processing*, 33:709–721, 2024. 3, 7