Exploring Salient Object Detection with Adder Neural Networks

Bo-Wen Yin[1], **Zheng Lin**[2*],

[1]VCIP, College of Computer Science, Nankai University
[2]BNRist, Department of Computer Science and Technology, Tsinghua University
bowenyin@mail.nankai.edu.cn, frazer.linzheng@gmail.com

Abstract

In this paper, we explore how to develop salient object detection models using adder neural networks (ANNs), which are more energy efficient than convolutional neural networks (CNNs), especially for real-world applications. Based on our empirical studies, we show that directly replacing the convolutions in CNN-based models with adder layers leads to a substantial loss of activations in the decoder part. This makes the feature maps learned in the decoder lack pattern diversity and hence results in a significant performance drop. To alleviate this issue, by investigating the statistics of the feature maps produced by adder layers, we introduce a simple yet effective differential merging strategy to augment the feature representations learned by adder layers and present a simple baseline for SOD using ANNs. Experiments on popular salient object detection benchmarks demonstrate that our proposed method with a simple feature pyramid network (FPN) architecture achieves comparable performance to previous state-of-the-art CNN-based models and consumes much less energy. We hope this work could facilitate the development of ANNs in binary segmentation tasks.

Introduction

Salient object detection (SOD), as one of the essential tasks in computer vision, has attracted tremendous interest from both academic and industrial communities over the past two decades. Because of the ability to detect the most salient and attention-grabbing objects in a scene (Cheng et al. 2014; Borji et al. 2019), SOD plays a significant role in intelligent cameras, content-aware image editing (Avidan and Shamir 2007; Achanta and Süsstrunk 2009; Pritch, Kav-Venaki, and Peleg 2009), visual tracking (Hong et al. 2015; Mahadevan and Vasconcelos 2009; Smeulders et al. 2013), segmentation (Sun 2024), etc. With the in-depth study of deep learning techniques (Wang et al. 2023; Cheng and Sun 2024; Liu et al. 2022, 2024), fully convolutional networks have dominated the area of salient object detection and have been the mainstream in recent years (Hou et al. 2019; Yin et al. 2024b; Zhang et al. 2018; Qin et al. 2019; Wu, Su, and Huang 2019; Wang et al. 2019). Despite the excellent performance, most of these models require massive computations and consume tremendous energy, making them difficult to be deployed into mobile or edge devices. Thus, how to reduce the computation and energy cost of popular salient object detection models has been drawing significant attention.

Our work also focuses on studying how to reduce the energy cost for popular salient object detection models. Different from most previous works that reduce the computations and energy cost via model compression (Bucilu, Caruana, and Niculescu-Mizil 2006; Cheng et al. 2017; Nan et al. 2019), knowledge distillation (Hinton, Vinyals, and Dean 2015; Romero et al. 2014; Tung and Mori 2019), or channel pruning (He, Zhang, and Sun 2017; Zhuang et al. 2018; Gao et al. 2019), we attempt to investigate building models based on energy-efficient adder neural networks (ANNs) (Chen et al. 2020). ANNs, as described in (Chen et al. 2020), replace the multiplication operations in convolutions with additions, significantly decreasing the energy cost. However, unlike CNN-based methods, whose design practice has been extensively explored, ANNs are new and the design principle needs special attention (Chen et al. 2021). Therefore, how to develop strong ANNs for salient object detection deserves deep study.

In this paper, we select a representative CNN-based model, named PoolNet (Liu et al. 2019), as our exemplar and explore how to make the adder layers suit popular CNN-based architectures. We empirically found that directly replacing the convolutions with the adder layers in the PoolNet (Liu et al. 2019) with an ImageNet pretrained backbone still leads to a significant performance drop. We visualize the decoder part and observe that only a small number of the neurons are activated, while most channels within the same layer share nearly the same pattern. Consequently, the diversity of the feature maps cannot be guaranteed, which is essential for building robust networks.

To overcome the aforementioned issues brought in by the adder layers in the decoder, we present a simple scheme for ANN-based salient object detection models. Our method introduces a differential merging strategy to augment the diversity of the output feature maps of the adder layers. Specifically, we first split an adder layer into two separate branches and then compute the differences between the two branches. This allows us to adjust the statistics of the output feature maps and avoid the case in which most of the feature values are negative in the decoder part. In addition, regarding the

*Corresponding Author.

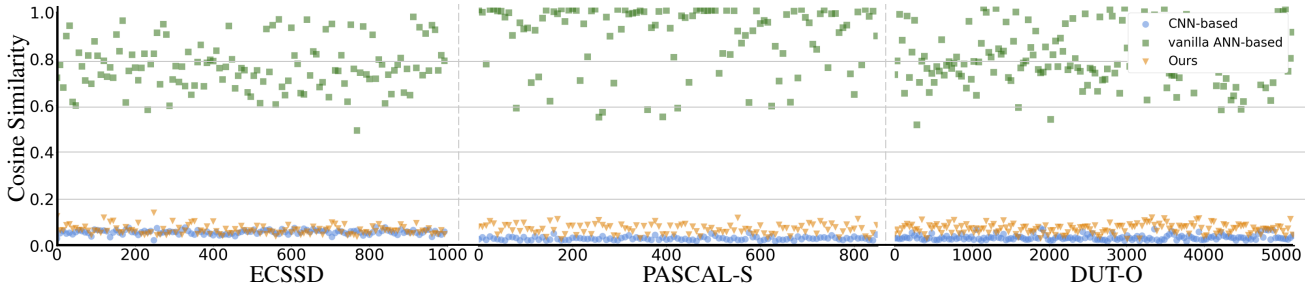


Figure 1: Average cosine similarity among feature maps of the same layer with different neural networks. We report results on the ECSSD (Yan et al. 2013), PASCAL-S (Li et al. 2014), and DUT-O (Yang et al. 2013) datasets. The horizontal axis represents the image serial number of each dataset. In the vertical axis, larger values mean that the similarity among channels is high.

Settings		ECSSD		DUT-O		PASCAL-S	
Encoder	Decoder	F↑	MAE ↓	F ↑	MAE ↓	F ↑	MAE ↓
Conv	Conv	**.933**	**.035**	**.772**	**.054**	**.832**	**.067**
Adder	Conv	.924	.037	.770	**.054**	.828	.068
Conv	Adder	.881	.060	.733	.065	.777	.075
Adder	Adder	.829	.088	.709	.077	.747	.084

Table 1: PoolNet performance on three SOD datasets (*e.g.* ECSSD (Yan et al. 2013), PASCAL-S (Li et al. 2014), and DUT-O (Yang et al. 2013)) when the encoder and decoder adopt different operators. 'Adder' and 'Conv' means we adopt the adder and convolutional operations in the encoder and decoder, respectively. '↑': the higher the better, 'F': short for max F-measure (Achanta et al. 2009), 'MAE': mean absolute error (Achanta et al. 2009).

fact that adder layers lack the ability to scale feature values and energy cost brought in by multiplications is higher than additions, we propose to assign each channel a weight factor to adjust the importance of each channel.

Our method is simple and easy to implement and can be regarded as a plug-and-play module. We will show that by using the simple AdderNet-50 backbone (Chen et al. 2020) plus the feature pyramid architecture (Lin et al. 2017), our method achieves comparable results to most previous state-of-the-art methods, like ICON (Zhuge et al. 2022) and EDN (Wu et al. 2022) on five widely used datasets. In the meantime, under the same architecture, our proposed ANN-based models saves around 55% of the energy consumption averagely compared to the CNN-based counterpart, providing a promising way to build efficient salient object detection models. In addition, we also study the impact of different basic training factors on ANN-based models.

To sum up, the main contributions of this paper can be summarized as follows:

- First of all, we conduct an in-depth study of ANNs for salient object detection and analyze how the basic training factors influence the model performance;
- Second, we present a simple yet effective differential merging strategy to compensate for the defect of the normal adder layers when applied to salient object detection;
- At last, we set up a new baseline for salient object detection based on ANNs. We show that our method with a simple feature pyramid network architecture achieves comparable performance to most previous state-of-the-art CNN-based models while costing much less energy.

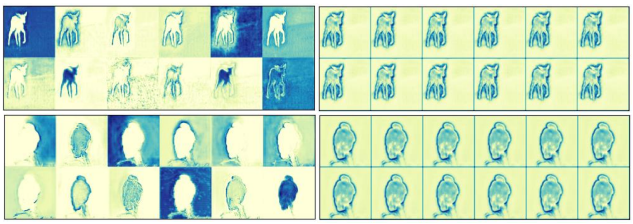
ANN Study for Salient Object Detection

Unlike convolutions that utilize cross-correlation to measure the similarity between the filters and the input features, AdderNets (Chen et al. 2020) adopt l_1-norm to compute the similarity between the filter weights and the input features. A straightforward way to apply AdderNets to SOD is to replace all the convolutional layers of a CNN-based model with the adder layers. Though the model composed of adder filters is efficient, one of the biggest problems is that the performance drops significantly compared to its CNN counterpart. Here, we first show the differences between CNNs and ANNs when applied to salient object detection and then study how the basic training factors affect performance.

ANN Study

We take one of the most representative models PoolNet (Liu et al. 2019) (without the edge detection part for simplicity) as our exemplar and study how the adder and convolutional layers perform under this architecture. The PoolNet architecture adopts an improved version of the feature pyramid network (Lin et al. 2017) as its decoder. Specifically, after merging the feature maps from two different levels, a feature aggregation module (FAM) that aggregates features from multiple scales is added. For more specific descriptions, readers can refer to the original paper (Liu et al. 2019).

Encoder and decoder analysis. To better understand how would the adder operations perform, we first separate the whole network into two parts: an encoder which corresponds to the backbone pretrained on the ImageNet dataset (Deng et al. 2009) and a decoder which corresponds to the rest layers. In Tab. 1, we show the results when the encoder and decoder adopt either the convolution or the adder operation based on the PoolNet architecture. We empirically found that when we replace the ResNet-50 backbone (He et al. 2016) with the AdderNet-50, the performance drops slightly, while when we keep the backbone unchanged and replace all the convolutions with adder layers in the decoder, the performance drops significantly. This implies that adder



CNN-based ANN-based

Figure 2: Visualization of features token from the CNN version PoolNet and the ANN version. As can be seen, the features from CNN-based PoolNet are much more diverse than those from the ANN-based PoolNet.

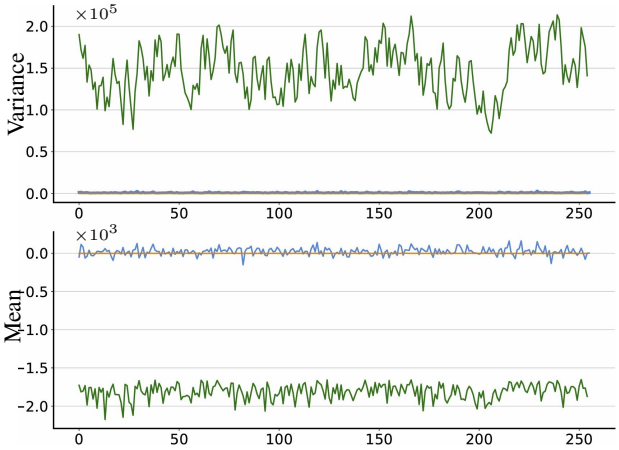
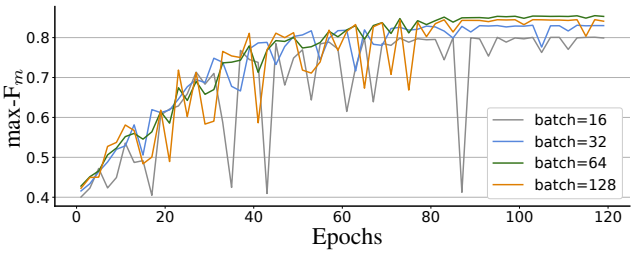


Figure 3: Statistics of the running mean and running variance of the batch normalization layer after the adder layer in the decoder of CNN-based PoolNet (orange line), ANN-based one (green line), and our method (blue line). We can see that directly using adder layers in the decoder makes the running mean and variance fluctuates erratically.

layers rely more on pretraining than CNNs, and directly taking the place of the convolutions with adder layers in CNN-based decoders is inappropriate.

Similarity analysis. To explore what leads to the performance drop, we use cosine similarity to quantitatively measure the differences among different channels produced by the same layer. As shown in Fig. 1, the cosine similarities of the feature maps generated by ANN are mainly distributed between 0.6 and 1, while the ones for CNN-based PoolNet are quite close to 0. This indicates that adder layers prevent the SOD model from learning expressive feature maps and hence lead to the performance drop. We also visualize the decoder feature maps of the CNN-based PoolNet and the ANN-based PoolNet (both encoder and decoder adopts adder layers) for a comparison. From Fig. 2, we can observe that the feature maps generated by the adder filters from the same layer share nearly the same pattern across channels, but the patterns of the original PoolNet are much more diverse.

Statistics of BN parameters. It has been shown in (Chen et al. 2021) that batch normalization (Ioffe and Szegedy 2015) plays an essential role in training ANN-based object detection models. Inspired by this observation, we further visualize the statistics of the parameters of the batch normalization layer in Fig. 3. We can see that the mean value for each channel in the decoder of the ANN-based PoolNet is much smaller than that of the CNN-based PoolNet, which means the utilization of the adder operation brings drastic variance to the output features. This makes ANN-based PoolNet difficult to be optimized and prevents the generated feature maps from being expressive, hence causing the performance drop. Therefore, how to adjust the statistics of the feature maps in the decoder is essential for adder layers to be applied to salient object detection models.



batch size	16	32	64	128
max-F_m	0.801	0.831	0.855	0.845

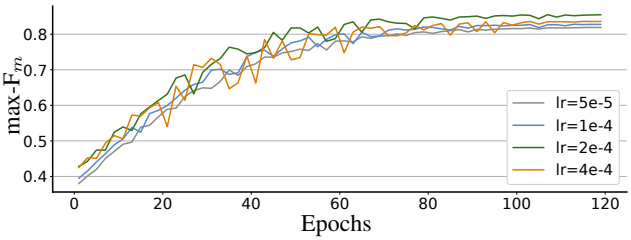
Figure 4: Performance curves of the Adder version PoolNet on DUTS-TE when using different batch sizes. We use cosine learning rate decay and the initial learning rate is 2e-4. We follow the curve style used in (Chen et al. 2021).

Influence of Basic Training Factors

Batch size. Batch normalization has a significant impact on the performance of ANN-based models. Batch size also influences the effect of batch normalization. In Fig. 4, we depict the performance curves using the Adder version PoolNet with different batch sizes. When the batch size is small (16), the fluctuation of the performance curve is considerable. Gradually increasing the batch size makes the training more stable, and the final performance also tends to improve. However, when the batch size is 128, the final performance drops. According to the above observation, we suggest setting the batch size to 64.

Learning rate. Fig. 5 shows the performance curves when different initial learning rates are used. We can see that increasing the learning rate from 5e-5 to 2e-4 can lift the performance. However, using a larger learning rate 4e-4 leads to a performance drop from 0.855 to 0.835 in terms of F-measure. In addition, a proper initial learning rate makes the training process more stable.

Training epochs. For CNN-based SOD models, the training epochs are often less than 60. For ANN-based ones, we found that a longer training procedure yields better results. Fig. 7 shows the performance curves when using different training epochs. We argue that the adder layers have no multiplication operations and hence converge slower than traditional convolution layers.



learning rate	5e-5	1e-4	2e-4	4e-4
max-F$_m$	0.819	0.828	0.855	0.836

Figure 5: Performance curves of the Adder version PoolNet on DUTS-TE when adopting different initial learning rates. The batch size is set to 64.

Method

As shown in Fig. 3, the running mean in the batch normalization layer is around -2×10^3. Since batch normalization is added after the adder layers, this reflects that the mean value for each channel of the adder layer output is not close to 0, making the ANN-based models difficult to converge well. To alleviate the above-mentioned problem of ANNs when applied to salient object detection, we present a simple yet effective differential merging module (DMM) by dynamically adjusting the statistics of feature maps. In addition, we also advance the classic channel attention mechanism to make it suited for ANNs.

Differential merging. Considering the above observation, we propose to split an adder layer into two separate branches and simply flip the signs of all the feature values in one branch. A visual illustration can be found on the right of Fig. 6. By simply computing the differences between the two branches, we can easily adjust the means and variances of the output feature maps to be near zero. This can be reflected by observing the orange curves depicted in Fig. 3. We can see that after introducing the differential merging module, the statistics of the batch normalization layers are quite similar to that in CNNs. In our experiment section, we will demonstrate the effectiveness of the proposed differential merging module using numerical results.

Differential channel attention. Though the DMM described above can effectively relieve the problem, the performance of our ANN-based model still lags behind its CNN-based counterpart. We argue that one of the main reasons is the absence of multiplication operations compared to convolutions, which can scale up or down the feature values, generating expressive feature patterns. To avoid the heavy use of multiplications, as a compromise, we propose to assign each channel an input-dependent factor to alleviate the scaling problem of adder layers. Our design is inspired by the widespread SE attention (Hu, Shen, and Sun 2018), but differently, we adopt the above differential merging strategy to learn the factor for each channel, which costs negligible energy consumption.

Our design, named differential channel attention (DCA), is more suitable for ANNs. An illustration can be found at the bottom right of Fig. 6. Taking into account the problem of adder layers, we apply the differential merging strategy to the first adder layer instead of simply replacing the convolutions in the original channel attention with adder operations. Our DCA can help further improve the model performance.

Setting	DUT-O		PASCAL-S		Params	Energy
	maxF ↑	MAE ↓	maxF ↑	MAE ↓		
ANN	0.647	0.084	0.764	0.107	64.2M	44.3mJ
ANN*	0.687	0.074	0.793	0.094	106.3M	70.3mJ
ANN + DMM	**0.725**	**0.064**	**0.813**	**0.076**	93.3M	63.6mJ

Table 2: Effect of DMM. 'ANN*': we increase the number of channels in the decoder of the ANN baseline to make a fair comparison. Simply increasing the number of channels in the decoder can improve the results but performs worse than our method.

Network Architecture. We utilize our differential merging module (DMM) with the differential channel attention (DCA) to build an efficient salient object detection architecture based on FPN (Lin et al. 2017). As shown in Fig. 6, our proposed network, namely DMNet, uses the AdderNet-50 (Chen et al. 2020) as the backbone and adopts an improved version of FPN (Lin et al. 2017). It is worth noting that the goal of this paper is to unveil that directly building an ANN-based salient object detection model by mimicking CNN-based models is not an appropriate way due to the intrinsic problems behind the adder operations. Meanwhile, we set a new baseline that would facilitate future work for salient object detection using ANNs.

In addition, we also modify some recent state-of-the-art CNN models to their ANN versions using the proposed DMM to test the versatility. Note that we do not change the network architecture but just replace the convolutions with the proposed DMM, followed by DCA.

Experimental Results

Experiment Setup

Implementation details. The implementation of the proposed method is based on the PyTorch (Paszke et al. 2019) framework. All the experiments are performed using the Adam (Kingma and Ba 2015) optimizer like (Yin et al. 2024a) with a batch size of 64. We train our network for 120 epochs as we found ANN-based models converge slower than CNN-based models. The learning rate is set to 2e-4 initially, and the cosine learning rate schedule is adopted. The backbone parameters are initialized with weights pretrained on the ImageNet-1K (Deng et al. 2009), and the parameters in the decoder are initialized randomly before training. We only use simple horizontal flipping for data augmentation.

Datasets & evaluation metrics. All the models are trained on the DUTS-TR dataset (Wang et al. 2017). To evaluate the performance of our proposed method, we conduct a series of experiments on five popular SOD datasets, *i.e.*, ECSSD (Yan et al. 2013), DUT-O (Yang et al. 2013), PASCAL-S (Li et al. 2014), HKU-IS (Li and Yu 2015), DUTS-TE (Wang et al.

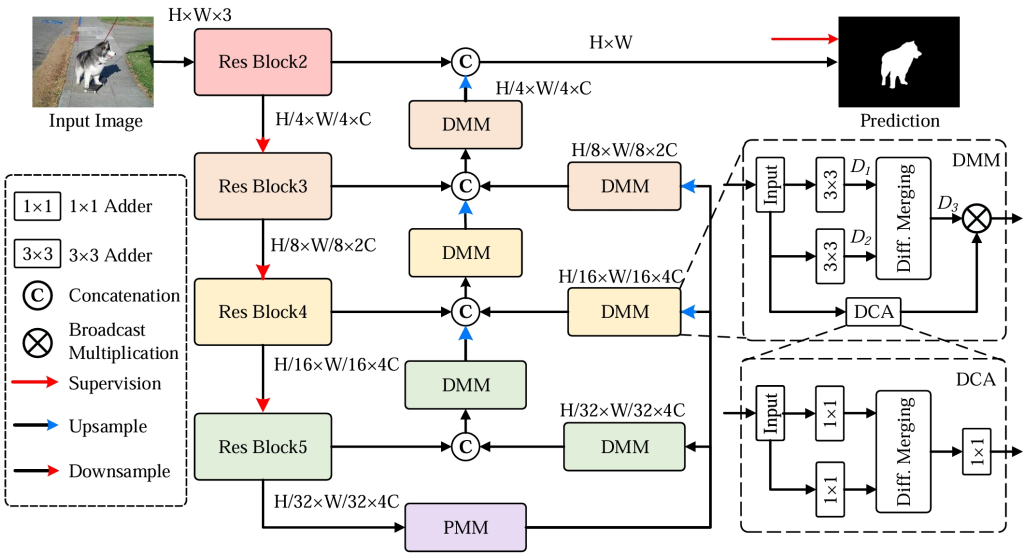
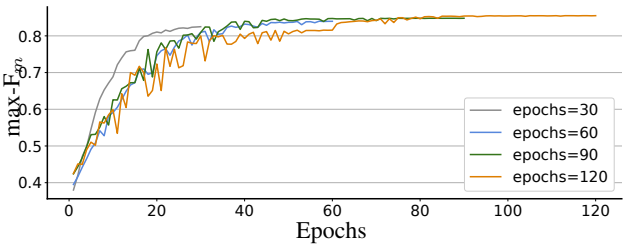


Figure 6: Illustration of the proposed DMNet. We omit the batch normalization and activation layers after each adder layer for simplicity. The global average pooling layers before the 1×1 Adder in DCA are also omitted. Similar to PoolNet (Liu et al. 2019), a simplified pyramid pooling module (Zhao et al. 2017) (PMM) is added to the top of the backbone. The backbone is Adder ResNet-50 (Chen et al. 2020).



training epochs	30	60	90	120
max-F_m	0.825	0.840	0.848	0.855

Figure 7: Performance curves of the Adder version PoolNet on DUTS-TE when training for different epochs. Clearly, a longer train time yields better results.

2017), containing 1000, 5168, 850, 4447, 5,019 pairs of images and ground-truth saliency maps, respectively.

Method Analysis

In this subsection, we provide a series of numerical analyses for the proposed method. We take the AdderNet-50 backbone with FPN plus PPM as our ANN baseline. In this subsection, we report results on two challenging datasets: DUT-O and PASCAL-S.

Differential merging. We first evaluate the performance of the proposed differential merging module. Note that in this experiment, differential channel attention is not used. Tab. 2 shows the result comparison. Equipped with our DMM, the ANN baseline shows consistent improvement on all the benchmarks. In Fig. 8, we visualize the prediction results. Using the proposed DMM can better identify the salient objects. To further exhibit the effect of our differential merging

Setting	DUT-O		PASCAL-S		Params	Energy
	maxF ↑	MAE ↓	maxF ↑	MAE ↓		
DMM	0.725	0.064	0.813	0.082	93.3M	63.6mJ
DMM + SE	0.735	0.061	0.817	0.080	93.6M	63.9mJ
DMM + Adder SE	0.731	0.064	0.815	0.083	93.6M	63.9mJ
DMM + DCA	**0.751**	**0.057**	**0.833**	**0.073**	93.7M	63.9mJ

Table 3: Effect of DCA. 'DMM baseline': our DMNet without DCA. For the experiments with the SE (Hu, Shen, and Sun 2018) module, we implement two different versions. We can see that DCA works much better than the SE module.

strategy, we also visualize the feature maps around DMM in Fig. 9. Before differential merging, the features D_1 and D_2 (marked in Fig. 6) are highly similar among channels. Differential merging generates diverse features which are close to the ones in CNN-based models, solving the problem of lacking diversity. Quantitatively, it also greatly reduces the performance drop when directly applying ANN.

Compared to the ANN baseline, our DMNet has more learnable parameters because two 3×3 adder layers are used in our DMM. To fairly show the effectiveness of the DMM, we also attempt to increase the channel number of the 3×3 adder layers in the decoder of the ANN baseline. As shown in Tab. 2, we can see that simply doubling the channel numbers of the 3×3 adder layers in the decoder can improve the results in terms of both maxF and MAE. However, when the DMM is used, the performance can be further boosted with even fewer learnable parameters and energy costs. This experiment further verifies the effectiveness of the proposed differential merging module.

Effect of DCA. We then evaluate the performance of our

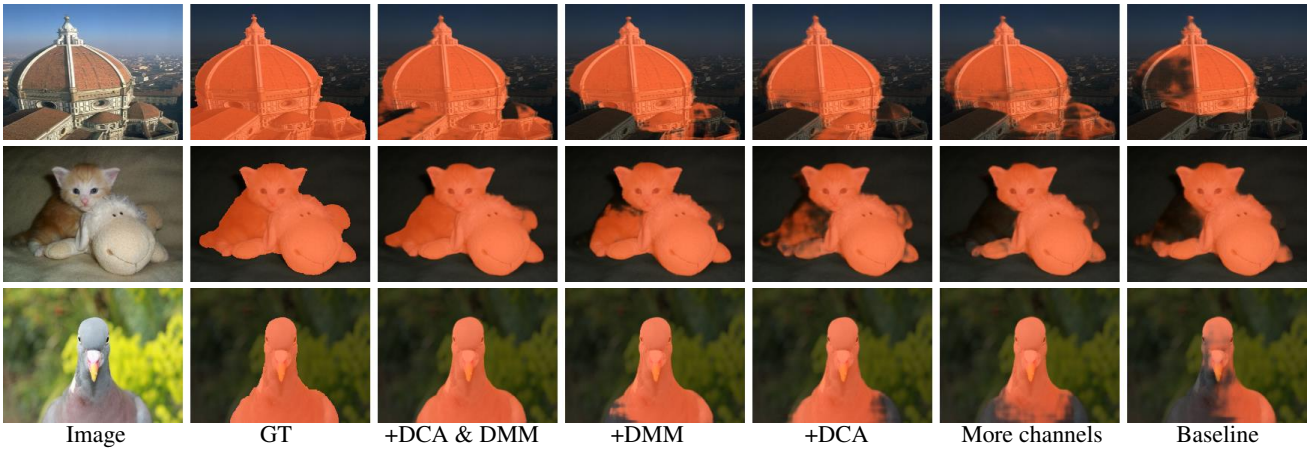


Figure 8: Visual comparisons using different combinations of our proposed differential merging module (DMM) and differential channel attention (DCA). From left to right: Source image; GT; +DCA & DMM (DMNet); ANN baseline + DMM; ANN baseline + DCA; ANN baseline with more channels; ANN baseline.

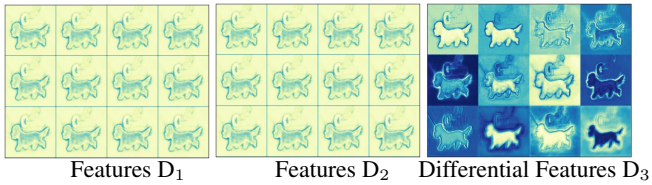


Figure 9: Visualization of features around DMM. We mark the positions of D_1 and D_2 in Fig. 6. Clearly, the differential features are much more diverse than the ones produced by the adder layers.

Settings			DUT-O		PASCAL-S		Params
ANN	DMM	DCA	maxF ↑	MAE ↓	maxF ↑	MAE ↓	
✓			0.647	0.084	0.764	0.107	64.2M
✓		✓	0.687	0.074	0.793	0.094	64.6M
✓	✓		0.725	0.064	0.813	0.082	93.3M
✓	✓	✓	**0.751**	**0.057**	**0.833**	**0.073**	93.7M

Table 4: Method analysis. 'DMM': different merging module without DCA; 'DCA': differential channel attention. All the experiments are based on the AdderNet-50 backbone (Chen et al. 2020). We can see that with the proposed DMM and DCA, the performance can be significantly improved on both the DUT-O and Pascal-S datasets.

differential channel attention (DCA). We first directly add the DCAs to the ANN baseline in the same way as in Fig. 6. Tab. 3 shows the result comparison. Equipping with DCA, the performance increases significantly, demonstrating that DCA also helps compared to the ANN baseline.

To further show the advantage of the proposed DCA over the classic channel attention mechanism in ANNs, we also attempt to replace the DCA in our DMNet with the SE module (Hu, Shen, and Sun 2018). We show two versions of the SE module: a convolutional version and an adder version. As shown in Tab. 3, compared to different versions of the SE module, our DCA yields better results. This indicates that our DCA is a more suitable channel attention mechanism for building ANN-based models compared to the classic SE module. We argue the main reason is the proposed differential merging module also helps in processing 1D features after global pooling.

DCA & DMM. Here, we test whether the proposed DCA is compatible with DMM. By comparing the last two rows of Tab. 4, it is obvious that introducing both DCA and DMM into our architecture further lifts the performance in terms of both the maxF and MAE scores. These results demonstrate that the two components are complementary. To further explain this, we also visualize some saliency results in Fig. 8. We can see that the complete version of our DMNet can not only locate the salient objects accurately but also generate high-quality prediction results.

Application to Other Models

Classic models. To further illustrate the generality of our proposed methods, we apply our DMM and DCA to other models for SOD. Firstly, We apply our DMM and DCA to the classic models, including FPN (Lin et al. 2017), PicaNet (Liu, Han, and Yang 2018), and PoolNet (Liu et al. 2019). For these three models, we only preserve three decoder blocks because it is enough for our method to guarantee performance. More specific implementations can be found in our supplementary material. The comparison of our methods and their corresponding CNN ones are shown in Tab. 5. To show the advantages of our ANN-based DMNet, we also follow prior works (Shu et al. 2021; Chen et al. 2021; Song et al. 2021) and compare the energy cost of different models. As can be seen, equipped with DMM and DCA, ANN-based models achieve comparable performance with their original versions but save around 70% on energy.

Recent SOTA models. Considering the classic methods are relatively simple on structures, we also adapt our design to SOTA models (e.g., EDN_{2022} (Wu et al. 2022) and

Model	Adder?		Consumption			PASCAL-S			ECSSD			DUTS-TE			DUT-O			HKU-IS		
	E	D	M (G)	A (G)	E (mJ)	M ↓	F↑	S ↑	M ↓	F ↑	S ↑	M ↓	F ↑	S ↑	M ↓	F ↑	S ↑	M ↓	F ↑	S ↑
FPN (Lin et al. 2017)			15.7	15.7	72.2	.081	.816	.845	.041	.917	.908	.043	.833	.858	.056	.752	.830	.035	904	.898
FPN ANN†	✓		9.3	21.5	53.8	.080	.820	.848	.044	.916	.904	.043	.835	.861	.058	.741	.825	.035	.909	.905
FPN ANN‡	✓	✓	0.9	27.4	28.0	.076	.830	.855	.045	.913	.902	.042	.838	.866	.059	.738	.822	.036	.907	.901
PicaNet (Liu, Han, and Yang 2018)			27.9	27.9	128.4	.078	.824	.852	.046	.926	.917	.051	.840	.869	.065	.761	.832	.043	.912	.904
PicaNet ANN†	✓		10.3	45.5	79.1	.076	.825	.852	.046	.928	.919	.047	.839	.865	.063	.751	.829	.038	.913	.907
PicaNet ANN‡	✓	✓	2.0	38.2	41.6	.075	.829	.850	.044	.924	.913	.040	.841	.870	.060	.740	.825	.035	.909	.902
PoolNet (Liu et al. 2019)			44.5	44.5	204.7	.067	.832	.864	.035	.933	.926	.037	.855	.887	.054	.772	.831	.030	.924	.919
PoolNet ANN†	✓		13.5	75.5	117.9	.069	.828	.857	.040	.925	.923	.039	.849	.883	.055	.761	.826	.033	.916	.915
PoolNet ANN‡	✓	✓	3.1	58.3	63.9	.071	.831	.860	.041	.921	.919	.038	.855	.884	.057	.757	.824	.033	.914	.911

Table 5: Qualtitive results of the proposed DMM on some classical SOD methods. As can be seen, our ANN-based models achieve comparable results to their corresponding CNN-based models but save much more energy cost. †: AdderNet-50 (Chen et al. 2020) backbone and CNN-based decoder, ‡: AdderNet-50 backbone and ANN-based decoder which composed by our DMM and DCA.

Model	Calculation Cost			PASCAL-S			ECSSD			DUTS-TE			DUT-O			HKU-IS		
	M (G)	A (G)	E (mJ)	M ↓	F↑	S ↑	M ↓	F ↑	S ↑	M ↓	F ↑	S ↑	M ↓	F ↑	S ↑	M ↓	F ↑	S ↑
CPD (Wu, Su, and Huang 2019)	17.8	17.8	81.9	.072	.864	.848	.037	.939	.918	.043	.865	.869	.056	.797	.825	.034	.925	.905
BASNet (Qin et al. 2019)	127.4	127.4	586.4	.076	.854	.838	.037	.942	.916	.047	.860	.866	.056	.805	.836	.032	.928	.909
AFNet (Feng, Lu, and Ding 2019)	21.7	21.7	99.8	.071	.868	.850	.042	.935	.914	.046	.862	.866	.057	.797	.826	.036	.923	.905
LDF (Wei et al. 2020)	31.2	31.2	143.5	.060	.855	.863	.034	.938	.924	.034	.877	.892	.052	.782	.839	.028	.929	.919
GateNet-R (Zhao et al. 2020)	162.1	162.1	745.7	.069	.883	.857	.040	.945	.920	.040	.888	.884	.055	.818	.837	.033	.933	.915
ITSD (Zhou et al. 2020)	57.5	57.5	264.5	.066	.870	.859	.035	.947	.925	.041	.882	.884	.061	.818	.880	.031	.934	.917
MINet (Pang et al. 2020)	137.0	137.0	630.2	.064	.882	.857	.033	.947	.925	.037	.884	.884	.055	.810	.833	.028	.935	.920
DCN (Wu, Su, and Huang 2021)	110.2	110.2	506.9	.062	.853	.861	.031	.943	.928	.035	.876	.892	.051	.789	.845	.027	.930	.922
CTDNet (Zhao et al. 2021)	24.7	24.7	113.6	.061	.858	.863	.032	.939	.925	.034	.881	.893	.052	.794	.844	.027	.932	.921
AMSF (Zhang et al. 2021)	49.0	49.0	225.4	.061	.849	.852	.033	.934	.914	.034	.863	.877	.050	.778	.832	.027	.921	.908
ICON (Zhuge et al. 2022)	24.9	24.9	114.5	.064	.860	.861	.032	.943	.929	.037	.877	.888	.057	.799	.844	.029	.930	.920
ICON ANN†	13.3	36.6	82.2	.066	.855	.857	.036	.930	.917	.038	.873	.885	.058	.793	.839	.032	.923	.913
ICON ANN‡	5.5	59.8	74.2	.066	.853	.858	.036	.931	.919	.040	.864	.880	.063	.780	.831	.031	.923	.914
EDN (Wu et al. 2022)	20.4	20.4	93.8	.062	.860	.865	.032	.941	.927	.035	.878	.892	.049	.799	.849	.027	.933	.924
EDN ANN†	8.8	32.2	61.5	.066	.851	.857	.036	.934	.921	.040	.864	.880	.054	.787	.840	.030	.929	.918
EDN ANN‡	3.4	47.4	55.2	.072	.840	.849	.040	.930	.918	.042	.864	.880	.055	.787	.839	.033	.926	.916
DMNet ANN‡	2.6	43.5	48.8	.073	.833	.868	.039	.923	.920	.042	.848	.879	.057	.751	.822	.031	.920	.925

Table 6: Quantitative comparisons of our methods on ICON (Zhuge et al. 2022) and EDN (Wu et al. 2022) with other SOTA SOD models. As can be seen, our ANN-based DMNet achieves comparable results to most previous CNN-based models but saves much more energy cost. '†': AdderNet-50 (Chen et al. 2020) backbone and CNN-based decoder, '‡': AdderNet-50 backbone and ANN-based decoder which is composed of our DMM and DCA.

ICON_{2022} (Zhuge et al. 2022)) to indicate the generality of our method on complex structures. As shown in Tab. 6, compared with CNN-based ICON (Zhuge et al. 2022) and EDN (Wu et al. 2022), ANN-based ones achieve comparable performance as 65% and 59% of energy are preserved. Though there is still a margin between our models and the state-of-the-art CNN-based models, regarding the low energy cost, ANN-based ones would be a good trade-off for salient object detection between accuracy performance and energy efficiency. Particularly worth mentioning is that the goal of this paper is to provide an initial empirical study of ANNs for salient object detection and set a baseline. We believe more complex architectures could further improve the results but exploring them is beyond the scope of this paper.

Conclusion

In this paper, we explore salient object detection using adder neural networks. We reveal that the feature maps generated by adder layers in SOD decoders are short of pattern diversity. By observing the intermediate feature maps and the statistics of the batch normalization layer, we found that a simple differential merging strategy can well remedy the defect of adder layers as mentioned above. To unleash the potential of our designs, we go back to basics and investigate the effects of batch size, learning rate, and training epochs on training ANN-based models. Extensive experiments on five benchmarks show that SOTA models equipped with our designs can achieve comparable performance to their CNN-based ones but save much more energy consumption.

Acknowledgments

This work is supported by the China Postdoctoral Science Foundation under Grant Number GZB20240357, 2024M761682 and Shui Mu Tsinghua Scholar under Grant Number 2024SM079.

References

Achanta, R.; Hemami, S.; Estrada, F.; and Susstrunk, S. 2009. Frequency-tuned salient region detection. In *IEEE CVPR*.

Achanta, R.; and Süsstrunk, S. 2009. Saliency detection for content-aware image resizing. In *IEEE ICIP*, 1005–1008.

Avidan, S.; and Shamir, A. 2007. Seam carving for content-aware image resizing. In *ACM SIGGRAPH*, 10–es.

Borji, A.; Cheng, M.-M.; Hou, Q.; Jiang, H.; and Li, J. 2019. Salient object detection: A survey. *CVM*, 5(2): 117–150.

Bucilu, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *ACM SIGKDD*, 535–541.

Chen, H.; Wang, Y.; Xu, C.; Shi, B.; Xu, C.; Tian, Q.; and Xu, C. 2020. AdderNet: Do we really need multiplications in deep learning? In *IEEE CVPR*, 1468–1477.

Chen, X.; Xu, C.; Dong, M.; Xu, C.; and Wang, Y. 2021. An empirical study of adder neural networks for object detection. *NeurIPS*, 34.

Cheng, M.-M.; Mitra, N. J.; Huang, X.; Torr, P. H.; and Hu, S.-M. 2014. Global contrast based salient region detection. *IEEE TPAMI*, 37(3): 569–582.

Cheng, S.; and Sun, H. 2024. SPT: Sequence Prompt Transformer for Interactive Image Segmentation. *arXiv preprint arXiv:2412.10224*.

Cheng, Y.; Wang, D.; Zhou, P.; and Zhang, T. 2017. A survey of model compression and acceleration for deep neural networks. *IEEE Signal Process Mag*.

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

Feng, M.; Lu, H.; and Ding, E. 2019. Attentive Feedback Network for Boundary-Aware Salient Object Detection. In *IEEE CVPR*.

Gao, X.; Zhao, Y.; Dudziak, Ł.; Mullins, R.; and Xu, C.-z. 2019. Dynamic channel pruning: Feature boosting and suppression. *ICLR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *IEEE CVPR*.

He, Y.; Zhang, X.; and Sun, J. 2017. Channel pruning for accelerating very deep neural networks. In *IEEE CVPR*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hong, S.; You, T.; Kwak, S.; and Han, B. 2015. Online tracking by learning discriminative saliency map with convolutional neural network. In *ICML*, 597–606.

Hou, Q.; Cheng, M.-M.; Hu, X.; Borji, A.; Tu, Z.; and Torr, P. H. 2019. Deeply Supervised Salient Object Detection with Short Connections. *IEEE TPAMI*, 41(04): 815–828.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *IEEE CVPR*.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*.

Kingma, D. P.; and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Li, G.; and Yu, Y. 2015. Visual saliency based on multiscale deep features. In *IEEE CVPR*.

Li, Y.; Hou, X.; Koch, C.; Rehg, J. M.; and Yuille, A. L. 2014. The secrets of salient object segmentation. In *IEEE CVPR*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *IEEE CVPR*.

Liu, J.-J.; Hou, Q.; Cheng, M.-M.; Feng, J.; and Jiang, J. 2019. A simple pooling-based design for real-time salient object detection. In *IEEE CVPR*, 3917–3926.

Liu, J.-J.; Hou, Q.; Liu, Z.-A.; and Cheng, M.-M. 2022. PoolNet+: Exploring the Potential of Pooling for Salient Object Detection. *IEEE TPAMI*.

Liu, N.; Han, J.; and Yang, M.-H. 2018. Picanet: Learning pixel-wise contextual attention for saliency detection. In *IEEE CVPR*, 3089–3098.

Liu, N.; Luo, Z.; Zhang, N.; and Han, J. 2024. Vst++: Efficient and stronger visual saliency transformer. *IEEE TPAMI*.

Mahadevan, V.; and Vasconcelos, N. 2009. Saliency-based discriminant tracking. In *IEEE CVPR*, 1007–1013.

Nan, K.; Liu, S.; Du, J.; and Liu, H. 2019. Deep model compression for mobile platforms: A survey. *Tsinghua Sci Technol*, 24(6): 677–693.

Pang, Y.; Zhao, X.; Zhang, L.; and Lu, H. 2020. Multi-scale interactive network for salient object detection. In *IEEE CVPR*, 9413–9422.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*.

Pritch, Y.; Kav-Venaki, E.; and Peleg, S. 2009. Shift-map image editing. In *IEEE ICCV*.

Qin, X.; Zhang, Z.; Huang, C.; Gao, C.; Dehghan, M.; and Jagersand, M. 2019. Basnet: Boundary-aware salient object detection. In *IEEE CVPR*.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Shu, H.; Wang, J.; Chen, H.; Li, L.; Yang, Y.; and Wang, Y. 2021. Adder Attention for Vision Transformer. In *NeurIPS*.

Smeulders, A. W.; Chu, D. M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; and Shah, M. 2013. Visual tracking: An experimental survey. *IEEE TPAMI*, 36(7): 1442–1468.

Song, D.; Wang, Y.; Chen, H.; Xu, C.; Xu, C.; and Tao, D. 2021. Addersr: Towards energy efficient image super-resolution. In *IEEE CVPR*, 15648–15657.

Sun, H. 2024. Ultra-High Resolution Segmentation via Boundary-Enhanced Patch-Merging Transformer. *arXiv preprint arXiv:2412.10181*.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *IEEE ICCV*.

Wang, L.; Lu, H.; Wang, Y.; Feng, M.; Wang, D.; Yin, B.; and Ruan, X. 2017. Learning to detect salient objects with image-level supervision. In *IEEE CVPR*.

Wang, W.; Zhao, S.; Shen, J.; Hoi, S. C.; and Borji, A. 2019. Salient object detection with pyramid attention and salient edges. In *IEEE CVPR*.

Wang, Y.; Wang, R.; Fan, X.; Wang, T.; and He, X. 2023. Pixels, regions, and objects: Multiple enhancement for salient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10031–10040.

Wei, J.; Wang, S.; Wu, Z.; Su, C.; Huang, Q.; and Tian, Q. 2020. Label Decoupling Framework for Salient Object Detection. In *IEEE CVPR*.

Wu, Y.-H.; Liu, Y.; Zhang, L.; Cheng, M.-M.; and Ren, B. 2022. EDN: Salient object detection via extremely-downsampled network. *IEEE TIP*, 31: 3125–3136.

Wu, Z.; Su, L.; and Huang, Q. 2019. Cascaded partial decoder for fast and accurate salient object detection. In *IEEE CVPR*.

Wu, Z.; Su, L.; and Huang, Q. 2021. Decomposition and completion network for salient object detection. *IEEE TIP*, 30: 6226–6239.

Yan, Q.; Xu, L.; Shi, J.; and Jia, J. 2013. Hierarchical saliency detection. In *IEEE CVPR*.

Yang, C.; Zhang, L.; Lu, H.; Ruan, X.; and Yang, M.-H. 2013. Saliency detection via graph-based manifold ranking. In *IEEE CVPR*.

Yin, B.; Zhang, X.; Hou, Q.; Sun, B.-Y.; Fan, D.-P.; and Van Gool, L. 2024a. Camoformer: Masked separable attention for camouflaged object detection. *IEEE TPAMI*.

Yin, B.; Zhang, X.; Li, Z.; Liu, L.; Cheng, M.-M.; and Hou, Q. 2024b. DFormer: Rethinking RGBD Representation Learning for Semantic Segmentation. In *ICLR*.

Zhang, M.; Liu, T.; Piao, Y.; Yao, S.; and Lu, H. 2021. Automsfnet: Search multi-scale fusion network for salient object detection. In *ACM MM*.

Zhang, X.; Wang, T.; Qi, J.; Lu, H.; and Wang, G. 2018. Progressive attention guided recurrent network for salient object detection. In *IEEE CVPR*, 714–722.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *IEEE CVPR*, 2881–2890.

Zhao, X.; Pang, Y.; Zhang, L.; Lu, H.; and Zhang, L. 2020. Suppress and balance: A simple gated network for salient object detection. In *ECCV*.

Zhao, Z.; Xia, C.; Xie, C.; and Li, J. 2021. Complementary trilateral decoder for fast and accurate salient object detection. In *ACM MM*.

Zhou, H.; Xie, X.; Lai, J.-H.; Chen, Z.; and Yang, L. 2020. Interactive two-stream decoder for accurate and fast saliency detection. In *IEEE CVPR*.

Zhuang, Z.; Tan, M.; Zhuang, B.; Liu, J.; Guo, Y.; Wu, Q.; Huang, J.; and Zhu, J. 2018. Discrimination-aware channel pruning for deep neural networks. In *NeurIPS*.

Zhuge, M.; Fan, D.-P.; Liu, N.; Zhang, D.; Xu, D.; and Shao, L. 2022. Salient object detection via integrity learning. *IEEE TPAMI*.