

论文发布日期：2019.8.8[ICCV]

ICCV 2019南开程明明组的工作

# 1. Introduction

- 出发点：基于FCN方法的显著性检测任务（分割也有是类似的）由于是像素级的判别，缺少结构信息，导致显著性目标检测的边界不够精确。
- 解决方案：引入边缘信息作为监督，将边缘信息和显著性目标检测任务共同学习，并且互相特征复用、优势互补，能够取得更好的效果。

## 2. Salient Edge Guidance Network

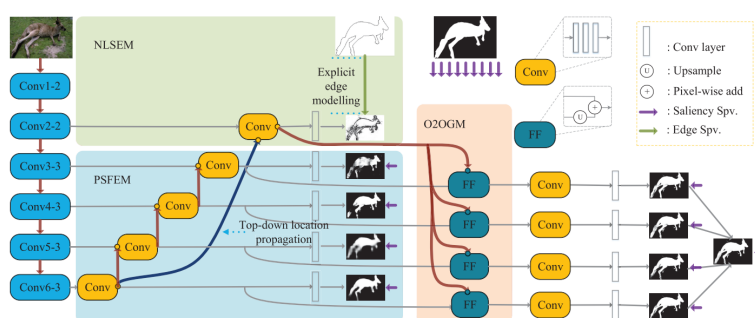


Figure 2. The pipeline of the proposed approach. We use brown thick lines to represent information flows between the scales. PSFEM: progressive salient object features extraction module. NLSEM: non-local salient edge features extraction module. O2OGM: one-to-one guidance module. FF: feature fusion. Spv.: supervision.

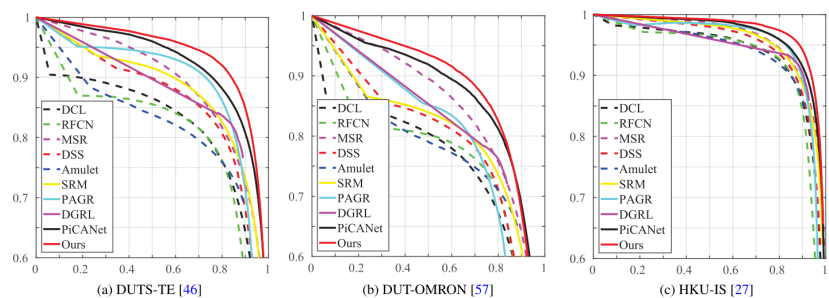
- **PSFEM**  
采用的backbone是U-Net，在提取不同尺度的特征图，并且通过类似FPN的结构将上层的语义信息向下传播；每一层特征图经过一层卷积后输出显著性检测结果；
- **NLSEM**  
加入边缘信息，底层融合了高层语义信息的特征图加入边缘信息学习边缘，用于后面的特征融合；
- **O2OGM**  
将NLSEM的卷积特征和PSFEM每个特征图的上采样结果进行像素相加，得到边缘和显著性检测的融合特征；该特征最终输出预测显著性目标结果。

# 3. Result

不太了解显著性检测，咱也不知道，咱也不好说，但是效果应该是拔萃的，直接和SOTA比了个遍，指标上全面碾压：

	ECSSD [56]			PASCAL-S [30]			DUT-O [57]			HKU-IS [27]			SOD [36,37]			DUTS-TE [46]		
	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑	MaxF ↑	MAE ↓	S ↑
VGG-based																		
DCL* [28]	0.896	0.080	0.863	0.805	0.115	0.791	0.733	0.094	0.743	0.893	0.063	0.859	0.831	0.131	0.748	0.786	0.081	0.785
DSS* [17,18]	0.906	0.064	0.882	0.821	0.101	0.796	0.760	0.074	0.765	0.900	0.050	0.878	0.834	0.125	0.744	0.813	0.065	0.812
MSR [26]	0.903	0.059	0.875	0.839	0.083	0.802	0.790	0.073	0.767	0.907	0.043	0.852	0.841	0.111	0.757	0.824	0.062	0.809
NLDF [35]	0.903	0.065	0.875	0.822	0.098	0.803	0.753	0.079	0.750	0.902	0.048	0.878	0.837	0.123	0.756	0.816	0.065	0.805
RAS [3]	0.915	0.060	0.886	0.830	0.102	0.798	0.784	0.063	0.792	0.910	0.047	0.884	0.844	0.130	0.760	0.800	0.060	0.827
ELD* [13]	0.865	0.082	0.839	0.772	0.122	0.757	0.738	0.093	0.743	0.843	0.072	0.823	0.762	0.154	0.705	0.747	0.092	0.749
DHS [32]	0.905	0.062	0.884	0.825	0.092	0.807	-	-	-	0.892	0.052	0.869	0.823	0.128	0.750	0.815	0.065	0.809
RFCN* [48]	0.898	0.097	0.852	0.827	0.118	0.799	0.747	0.094	0.752	0.895	0.079	0.860	0.805	0.161	0.730	0.786	0.090	0.784
UCF [62]	0.908	0.080	0.884	0.820	0.127	0.806	0.735	0.131	0.748	0.888	0.073	0.874	0.798	0.164	0.762	0.771	0.116	0.777
Amulet [61]	0.911	0.062	0.894	0.826	0.092	0.820	0.737	0.083	0.771	0.889	0.052	0.886	0.799	0.146	0.753	0.773	0.075	0.796
C2S [29]	0.909	0.057	0.891	0.845	0.081	0.839	0.759	0.072	0.783	0.897	0.047	0.886	0.821	0.122	0.763	0.811	0.062	0.822
PAGR [63]	0.924	0.064	0.889	0.847	0.089	0.818	0.771	0.071	0.751	0.919	0.047	0.889	0.841	0.146	0.716	0.854	0.055	0.825
Ours	0.941	0.044	0.913	0.863	0.076	0.848	0.826	0.056	0.813	0.929	0.034	0.910	0.869	0.110	0.788	0.880	0.043	0.866
ResNet-based																		
SRM* [49]	0.916	0.056	0.895	0.838	0.084	0.832	0.769	0.069	0.777	0.906	0.046	0.887	0.840	0.126	0.742	0.826	0.058	0.824
DGRL [52]	0.921	0.043	0.906	0.844	0.075	0.839	0.774	0.062	0.791	0.910	0.036	0.896	0.843	0.103	0.774	0.828	0.049	0.836
PiCANet* [33]	0.932	0.048	0.914	0.864	0.077	0.850	0.820	0.064	0.808	0.920	0.044	0.905	0.861	0.103	0.790	0.863	0.050	0.850
Ours	0.943	0.041	0.918	0.869	0.074	0.852	0.842	0.052	0.818	0.937	0.031	0.918	0.890	0.097	0.807	0.893	0.039	0.875

PR曲线在三个数据集上也是凌驾于其他检测器，高高在上，十分嚣张的样子：



比较有意思的是引入的方式和融合中相关问题的讨论。纯粹引入边缘信息并不一定能带来很好的效果，这在目标检测中已经有很多工作做过了，从结果来看效果不怎么突出，远不像能取得这里很亮眼的表现。这里能work可能归因于他的一些融合考量和学习方法，加上边缘和显著性检测两个任务本身相似性很高，更符合MTL的方式。