

C⁴Net Supplementary Material and Ablation Study

BMVC 2021 Submission # 1077

0.1 Ablation Study

Features Combination Methods. Modern architectures of deep learning solutions for the salient object detection problem are based on encoders, decoders, and shortcut connections between them. Each layer of the decoder is responsible for combining features from the deeper layer of the decoder and the corresponding layer of the encoder. Some proposed solutions like [1, 2] even get additional feature representations from deep layers as global guiding information. There are two main approaches to use that information. The first one is by simply concatenating them, the second one is to fuse them by using other functions like multiplication or addition. Another very important thing is the features processing structure at every layer of the decoder. Methods like [1, 2] prefer to process encoder's and decoder's features separately and then fuse them by using multiplication.

We tried to find out the answers to two main questions

- How the features of the encoder and decoder need to be processed. (joint or separated)
- How they need to be combined. (fusing or concatenating)

To answer these questions, we proposed two main structures of a decoder's layer Figure 1. We designed a PipeMode layer, which is a joint processing of two feature representations, and BranchedMode layer, which is separated processing of the features. They both contain two aggregation functions R_1 and R_2 .

The corresponding features of encoding layers, $f_l^{(i)}$ contain low-level information like edges, tiny areas, and high-frequency data, which is crucial for high-quality detection, especially on edges and they contain a lot of noisy information. The feature representations of decoding layers, $f_h^{(i)}$ contain noisy free high-level information like class, position, or shape of the object.

These features representations helped Jun Wei *et al.* [3] to propose a solution like our BranchedMode, where they used multiplication function to fuse high and low-level features and to clean noisy parts, then they added a skip-like connection as complementary information. We find this approach has some drawbacks, because of the choice of the structure and aggregating functions, so we assume

- As shown in work [3], feature values become smaller in deeper layers, the fused feature values can be pushed to zero, because of the multiplication aggregation function and the counteracting weights usually initialized randomly around zero. As the feature

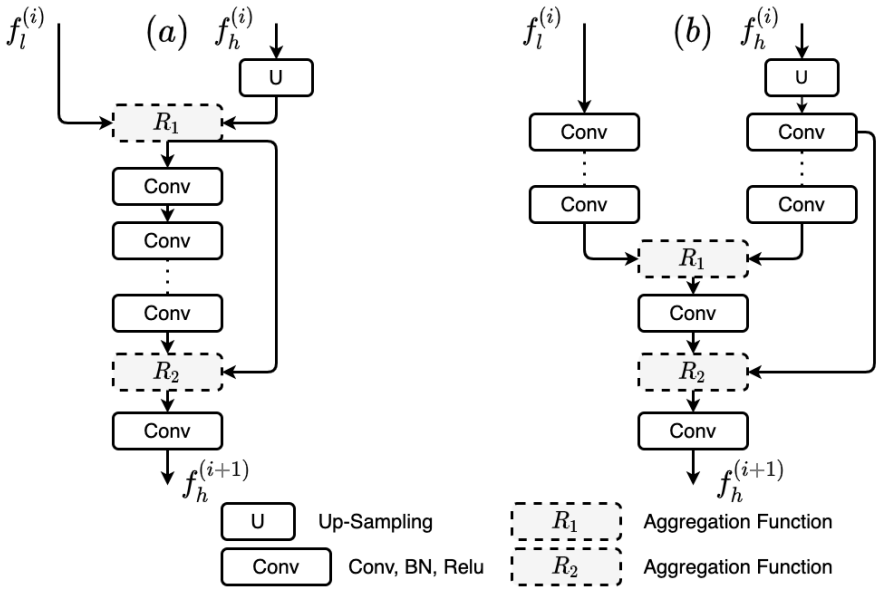


Figure 1: An overview of our proposed structures for a layer of a decoder. (a) is the joint-features module and we called it PipeMode and (b) is the separated-features module and we called it BranchedMode. R_1 and R_2 are aggregation functions

values are close to zero, the gradients also are close to zero and the network will learn slowly or will not learn at all.

- Based on our setup, the joint learning ((a) in Figure 1) gives better results than separated learning ((b) in Figure 1), because of the sharing information during feature representation processing.

To find out the real behavior of these two approaches, we made different experiments with these structures by using different aggregating functions. We designed a regular and simple architecture for semantic segmentation, which consists of a *ResNet50* encoder, our proposed two structures as the decoder's layers, and shortcut connections between them. The *binary cross-entropy* loss function was used on top of the first layer of the decoder. Each model was trained three times with randomly initialized weights and the result was calculated by taking the average of the best results at each run. In Table 1 we show the results of our experiments. Based on those results and the architecture choice, we can say:

- In general, PipeMode gives better results than BranchedMode, which shows that the sharing of information about different features leads to better results.
- The concatenation aggregation function works better for both structures.

PipeMM results are missing in Table 1, because it disturbs the training of the model with activation values pushed to zero [2].

Name	R_1	R_2	MAE	mF	E_{ξ}
BranchPP	Plus	Plus	0.0356	0.8514	0.9162
BranchMM	Mul	Mul	0.0358	0.8496	0.9165
BranchCC	Cat	Cat	0.0353	0.8535	0.9168
BranchMP	Mul	Plus	0.0353	0.8509	0.9152
PipePP	Plus	Plus	0.0348	0.8528	0.9164
PipeMM	Mul	Mul	-	-	-
PipeCC	Cat	Cat	0.0342	0.8512	0.9171
PipeCP	Cat	Plus	0.0347	0.8520	0.9159

Table 1: Results of our proposed structures on *DUTS-Test* dataset with different aggregating functions, where Plus is addition, Mul is multiplication and Cat is concatenation. The **green** is the overall best result, **red** is the overall worst result, **blue** is the best result among BranchedModes, **orange** is the best result among PipeModes.

References

[1] Zuyao Chen, Qianqian Xu, Runmin Cong, and Qingming Huang. Global context-aware progressive aggregation network for salient object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[2] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

[3] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[4] Jun Wei, Shuhui Wang, and Qingming Huang. F3net: Fusion, feedback and focus for salient object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.