IEEE International Conference on Computer Vision (ICCV) 2019

Optimizing the F-measure for Threshold-free Salient Object Detection

Kai Zhao[1], Shanghua Gao[1], Wenguan Wang[2], Ming-Ming Cheng[1]*
[1]TKLNDST, CS, Nankai University [2]Inception Institute of Artificial Intelligence
{kaiz.xyz,shanghuagao,wenguanwang.ai}@gmail.com,cmm@nankai.edu.cn

Abstract

Current CNN-based solutions to salient object detection (SOD) mainly rely on the optimization of cross-entropy loss (CELoss). Then the quality of detected saliency maps is often evaluated in terms of F-measure. In this paper, we investigate an interesting issue: can we consistently use the F-measure formulation in both training and evaluation for SOD? By reformulating the standard F-measure, we propose the relaxed F-measure *which is differentiable w.r.t the posterior and can be easily appended to the back of CNNs as the loss function. Compared to the conventional cross-entropy loss of which the gradients decrease dramatically in the saturated area, our loss function, named FLoss, holds considerable gradients even when the activation approaches the target. Consequently, the FLoss can continuously force the network to produce polarized activations. Comprehensive benchmarks on several popular datasets show that FLoss outperforms the state-of-the-art with a considerable margin. More specifically, due to the polarized predictions, our method is able to obtain high-quality saliency maps without carefully tuning the optimal threshold, showing significant advantages in real-world applications. Code and pretrained models are available at* `http://kaizhao.net/fmeasure`.

1. Introduction

We consider the task of salient object detection (SOD), where each pixel of a given image has to be classified as salient (outstanding) or not. The human visual system is able to perceive and process visual signals distinctively: interested regions are conceived and analyzed with high priority while other regions draw less attention. This capacity has been long studied in the computer vision community in the name of 'salient object detection', since it can ease the procedure of scene understanding [4]. The performance of modern salient object detection methods is often evaluated in terms of F-measure. Rooted from information retrieval [29], the F-measure is widely used as an evaluation metric in tasks where elements of a specified class have to be retrieved, especially when the relevant class is rare. Given the per-pixel prediction $\hat{Y}(\hat{y}_i \in [0,1], i=1,...,|Y|)$ and the ground-truth saliency map $Y(y_i \in \{0,1\}, i=1,...,|Y|)$, a threshold t is applied to obtain the binarized prediction $\dot{Y}^t(\dot{y}^t_i \in \{0,1\}, i=1,...,|Y|)$. The F-measure is then defined as the harmonic mean of precision and recall:

$$F(Y,\dot{Y}^t) = (1+\beta^2)\frac{\text{precision}(Y,\dot{Y}^t)\cdot\text{recall}(Y,\dot{Y}^t)}{\beta^2\text{precision}(Y,\dot{Y}^t)+\text{recall}(Y,\dot{Y}^t)}, \quad (1)$$

where $\beta^2 > 0$ is a balance factor between precision and recall. When $\beta^2 > 1$, the F-measure is biased in favour of recall and otherwise in favour of precision.

Most CNN-based solutions for SOD [11, 16, 30, 9, 31, 39, 33] mainly rely on the optimization of *cross-entropy loss* (CELoss) in an FCN [22] architecture, and the quality of saliency maps is often assessed by the F-measure. Optimizing the pixel-independent CELoss can be regarded as minimizing the mean absolute error (MAE=$\frac{1}{N}\sum_i^N|\hat{y}_i - y_i|$), because in both circumstances each prediction/ground-truth pair works independently and contributes to the final score equally. If the data labels have biased distribution, models trained with CELoss would make biased predictions towards the majority class. Therefore, SOD models trained with CELoss hold biased prior and tend to predict unknown pixels as the background, consequently leading to low-recall detections. The F-measure [29] is a more sophisticated and comprehensive evaluation metric which combines precision and recall into a single score and automatically offsets the unbalance between positive/negative samples.

In this paper, we provide a uniform formulation in both training and evaluation for SOD. By directly taking the evaluation metric, *i.e.* the F-measure, as the optimization target, we perform F-measure maximizing in an end-to-end manner. To perform end-to-end learning, we propose the *relaxed F-measure* to overcome the in-differentiability in the standard F-measure formulation. The proposed loss function, named FLoss, is decomposable w.r.t the posterior $\hat{Y}$ and thus can be appended to the back of a CNN as supervision without effort. We test the FLoss on several state-of-the-art SOD architectures and witness a visible performance

[*] M.M. Cheng is the corresponding author.

gain. Furthermore, the proposed FLoss holds considerable gradients even in the saturated area, resulting in polarized predictions that are stable against the threshold. Our proposed FLoss enjoys three favorable properties:

- Threshold-free salient object detection. Models trained with FLoss produce contrastive saliency maps in which the foreground and background are clearly separated. Therefore, FLoss can achieve high performance under a wide range of threshold.
- Being able to deal with unbalanced data. Defined as the harmonic mean of precision and recall, the F-measure is able to establish a balance between samples of different classes. We experimentally evidence that our method can find a better compromise between precision and recall.
- Fast convergence. Our method quickly learns to focus on salient object areas after only hundreds of iterations, showing fast convergence speed.

2. Related Work

We review several CNN-based architectures for SOD and the literature related to F-measure optimization.

Salient Object Detection (SOD). The convolutional neural network (CNN) is proven to be dominant in many subareas of computer vision. Significant progress has been achieved since the presence of CNN in SOD. The DHS net [19] is one of the pioneers of using CNN for SOD. DHS firstly produces a coarse saliency map with global cues, including contrast, objectness *et al.* Then the coarse map is progressively refined with a hierarchical recurrent CNN. The emergence of the fully convolutional network (FCN) [22] provides an elegant way to perform the end-to-end pixel-wise inference. DCL [16] uses a two-stream architecture to process contrast information in both pixel and patch levels. The FCN-based sub-stream produces a saliency map with pixel-wise accuracy, and the other network stream performs inference on each object segment. Finally, a fully connected CRF [14] is used to combine the pixel-level and segment-level semantics.

Rooted from the HED [34] for edge detection, aggregating multi-scale side-outputs is proven to be effective in refining dense predictions especially when the detailed local structures are required to be preserved. In the HED-like architectures, deeper side-outputs capture rich semantics and shallower side-outputs contain high-resolution details. Combining these representations of different levels will lead to significant performance improvements. DSS [11] introduces deep-to-shallow short connections across different side-outputs to refine the shallow side-outputs with deep semantic features. The deep-to-shallow short connections enable the shallow side-outputs to distinguish real salient objects from the background and meanwhile retain the high resolution. Liu *et al.* [18] design a pooling-based module to efficiently fuse convolutional features from a top-down pathway. The idea of imposing top-down refinement has also been adopted in Amulet [38], and enhanced by Zhao *et al.* [40] with bi-directional refinement. Later, Wang *et al.* [32] propose a visual attention-driven model that bridges the gap between SOD and eye fixation prediction. These methods mentioned above tried to refine SOD by introducing a more powerful network architecture, from recurrent refining network to multi-scale side-output fusing. We refer the readers to a recent survey [3] for more details.

F-measure Optimization. Despite having been utilized as a common performance metric in many application domains, optimizing the F-measure doesn't draw much attention until very recently. The works aiming at optimizing the F-measure can be divided into two subcategories [6]: (a) structured loss minimization methods such as [24, 25] which optimize the F-measure as the target during training; and (b) plug-in rule approaches which optimize the F-measure during inference phase [13, 7, 26, 37].

Much of the attention has been drawn to the study of the latter subcategory: finding an optimal threshold value which leads to a maximal F-measure given predicted posterior $\hat{Y}$. There are few articles about optimizing the F-measure during the training phase. Petterson *et al.* [24] optimize the F-measure indirectly by maximizing a loss function associated to the F-measure. Then in their successive work [25] they construct an upper bound of the discrete F-measure and then maximize the F-measure by optimizing its upper bound. These previous studies either work as post-processing, or are in-differentiable w.r.t posteriors, making them hard to be applied to the deep learning framework.

3. Optimizing the F-measure for SOD

3.1. The Relaxed F-measure

In the standard F-measure, the true positive, false positive and false negative are defined as the number of corresponding samples:

$$\begin{aligned} TP(\dot{Y}^t, Y) &= \sum\nolimits_i 1(y_i == 1 \text{ and } \dot{y}_i^t == 1), \\ FP(\dot{Y}^t, Y) &= \sum\nolimits_i 1(y_i == 0 \text{ and } \dot{y}_i^t == 1), \\ FN(\dot{Y}^t, Y) &= \sum\nolimits_i 1(y_i == 1 \text{ and } \dot{y}_i^t == 0), \end{aligned} \tag{2}$$

where Y is the ground-truth, $\dot{Y}^t$ is the binary prediction binarized by threshold t and Y is the ground-truth saliency map. $1(\cdot)$ is an indicator function that evaluates to 1 if its argument is true and 0 otherwise.

To incorporate the F-measure into CNN and optimize it in an end-to-end manner, we define a decomposable F-measure that is differentiable over posterior $\hat{Y}$. Based on this motivation, we reformulate the true positive, false positive and false negative based on the continuous posterior $\hat{Y}$:

$$\begin{aligned} TP(\hat{Y},Y) &= \sum\nolimits_i \hat{y}_i \cdot y_i, \\ FP(\hat{Y},Y) &= \sum\nolimits_i \hat{y}_i \cdot (1-y_i), \\ FN(\hat{Y},Y) &= \sum\nolimits_i (1-\hat{y}_i) \cdot y_i\ . \end{aligned} \tag{3}$$

Given the definitions in Eq. 3, precision p and recall r are:

$$p(\hat{Y},Y) = \frac{TP}{TP+FP}, \quad r(\hat{Y},Y) = \frac{TP}{TP+FN}. \tag{4}$$

Finally, our *relaxed F-measure* can be written as:

$$\begin{aligned} F(\hat{Y},Y) &= \frac{(1+\beta^2)p\cdot r}{\beta^2 p + r}, \\ &= \frac{(1+\beta^2)TP}{\beta^2(TP+FN)+(TP+FP)}, \\ &= \frac{(1+\beta^2)TP}{H}, \end{aligned} \tag{5}$$

where $H = \beta^2(TP+FN)+(TP+FP)$. Due to the relaxation in Eq. 3, Eq. 5 is decomposable w.r.t the posterior $\hat{Y}$, therefore can be integrated in CNN architecture trained with back-prop.

3.2. Maximizing F-measure in CNNs

In order to maximize the *relaxed F-measure* in CNNs in an end-to-end manner, we define our proposed F-measure based loss (FLoss) function $\mathcal{L}_F$ as:

$$\mathcal{L}_F(\hat{Y},Y) = 1 - F = 1 - \frac{(1+\beta^2)TP}{H}. \tag{6}$$

Minimizing $\mathcal{L}_F(\hat{Y},Y)$ is equivalent to maximizing the *relaxed F-measure*. Note again that $\mathcal{L}_F$ is calculated directly from the raw prediction $\hat{Y}$ without thresholding. Therefore, $\mathcal{L}_F$ is differentiable over the prediction $\hat{Y}$ and can be plugged into CNNs. The partial derivative of loss $\mathcal{L}_F$ over network activation $\hat{Y}$ at location i is:

$$\begin{aligned} \frac{\partial \mathcal{L}_F}{\partial \hat{y}_i} &= -\frac{\partial F}{\partial \hat{y}_i} \\ &= -\Big(\frac{\partial F}{\partial TP}\cdot\frac{\partial TP}{\partial \hat{y}_i} + \frac{\partial F}{\partial H}\cdot\frac{\partial H}{\partial \hat{y}_i}\Big) \\ &= -\Big(\frac{(1+\beta^2)y_i}{H} - \frac{(1+\beta^2)TP}{H^2}\Big) \\ &= \frac{(1+\beta^2)TP}{H^2} - \frac{(1+\beta^2)y_i}{H}. \end{aligned} \tag{7}$$

There is another alternative to Eq. 6 which maximize the log-likelihood of F-measure:

$$\mathcal{L}_{\log F}(\hat{Y},Y) = -\log(F), \tag{8}$$

and the corresponding gradient is

$$\frac{\partial \mathcal{L}_{\log F}}{\partial \hat{y}_i} = \frac{1}{F}\left[\frac{(1+\beta^2)TP}{H^2} - \frac{(1+\beta^2)y_i}{H}\right]. \tag{9}$$

We will theoretically and experimentally analyze the advantage of FLoss against Log-FLoss and CELoss in terms of producing polarized and high-contrast saliency maps.

3.3. FLoss vs Cross-entropy Loss

To demonstrate the superiority of our FLoss over the alternative Log-FLoss and the *cross-entropy loss* (CELoss), we compare the definition, gradient and surface plots of these three loss functions. The definition of CELoss is:

$$\mathcal{L}_{CE}(\hat{Y},Y) = -\sum\nolimits_i^{|Y|}\left(y_i\log\hat{y}_i + (1-y_i)\log(1-\hat{y}_i)\right), \tag{10}$$

where i is the spatial location of the input image and $|Y|$ is the number of pixels of the input image. The gradient of $\mathcal{L}_{CE}$ w.r.t prediction $\hat{y}_i$ is:

$$\frac{\partial \mathcal{L}_{CE}}{\partial \hat{y}_i} = \frac{y_i}{\hat{y}_i} - \frac{1-y_i}{1-\hat{y}_i}. \tag{11}$$

As revealed in Eq. 7 and Eq. 11, the gradient of CELoss $\frac{\partial \mathcal{L}_{CE}}{\partial \hat{y}_i}$ relies only on the prediction/ground-truth of a single pixel i; whereas in FLoss $\frac{\partial \mathcal{L}_F}{\partial \hat{y}_i}$ is globally determined by the prediction and ground-truth of ALL pixels in the image. We further compare the surface plots of FLoss, Log-FLoss and CELoss in a two points binary classification problem. The results are in Fig. 1. The two spatial axes represent the prediction $\hat{y}_0$ and $\hat{y}_1$, and the z axis indicates the loss value.

As shown in Fig. 1, the gradient of FLoss is different from that of CELoss and Log-FLoss in two aspects: (1) Limited gradient: the FLoss holds limited gradient values even the predictions are far away from the ground-truth. This is crucial for CNN training because it prevents the notorious gradient explosion problem. Consequently, FLoss allows larger learning rates in the training phase, as evidenced by our experiments. (2) Considerable gradients in the saturated area: in CELoss, the gradient decays when the prediction gets closer to the ground-truth, while FLoss holds considerable gradients even in the saturated area. This will force the network to have polarized predictions. Salient detection examples in Fig. 3 illustrate the 'high-contrast' and polarized predictions.

4. Experiments and Analysis

4.1. Experimental Configurations

Dataset and data augmentation. We uniformly train our model and competitors on the MSRA-B [20] training set for a fair comparison. The MSRA-B dataset with 5000 images in total is equally split into training/testing subsets.

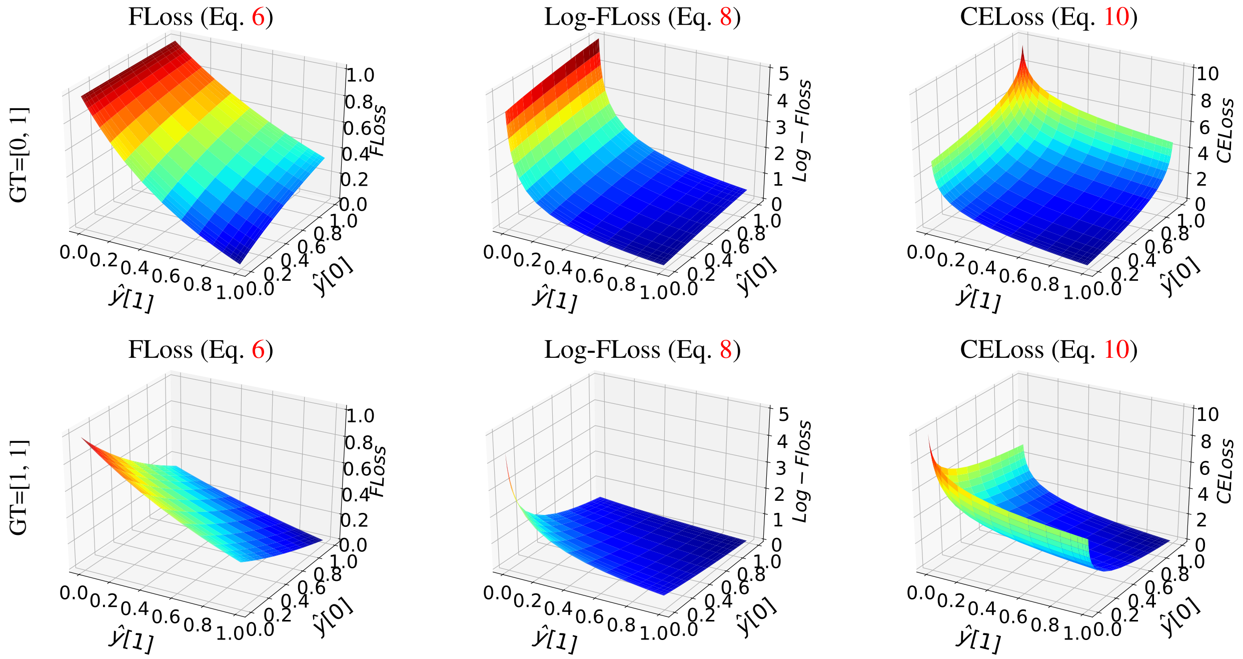


Figure 1. Surface plot of different loss functions in a 2-point 2-class classification circumstance. Columns from left to right: F-measure loss defined in Eq. 6, log F-measure loss defined in Eq. 8 and cross-entropy loss in Eq. 10. In top row the ground-truth is [0, 1] and in bottom row the ground-truth is [1, 1]. Compared with cross-entropy loss and Log-FLoss, FLoss holds considerable gradient even in the saturated area, which will force to produce polarized predictions.

We test the trained models on 5 other SOD datasets: ECSSD [35], HKU-IS [15], PASCALS [17], SOD [23], and DUT-OMRON [23]. More statistics of these datasets are shown in Table 1. It's worth mentioning that the challenging degree of a dataset is determined by many factors such as the number of images, number of objects in one image, the contrast of salient object w.r.t the background, the complexity of salient object structures, the center bias of salient objects and the size variance of images *etc*. Analyzing these details is out of the scope of this paper, we refer the readers to [8] for more analysis of datasets.

Dataset	#Images	Year	Pub.	Contrast
MSRA-B [20]	5000	2011	TPAMI	High
ECSSD [35]	1000	2013	CVPR	High
HKU-IS [15]	1447	2015	CVPR	Low
PASCALS [17]	850	2014	CVPR	Medium
SOD [23]	300	2010	CVPRW	Low
DUT-OMRON [36]	5168	2013	CVPR	Low

Table 1. Statistics of SOD datasets. '#Images' indicates the number of images in a dataset and 'contrast' represents the general contrast between foreground/background. The lower the contrast, the more challenging the dataset is.

Data augmentation is critical to generating sufficient data for training deep CNNs. We fairly perform data augmentation for the original implementations and their FLoss variants. For the DSS [11] and DHS [19] architectures we perform only horizontal flip on both training images and saliency maps just as DSS did. Amulet [38] only allows 256×256 inputs. We randomly crop/pad the original data to get square images, then resize them to meet the shape requirement.

Network architecture and hyper-parameters. We test our proposed FLoss on 3 baseline methods: Amulet [38], DHS [20] and DSS [11]. To verify the effectiveness of FLoss (Eq. 6), we replace the loss functions of the original implementations with FLoss and keep all other configurations unchanged. As explained in Sec. 3.3, the FLoss allows a larger base learning rate due to limited gradients. We use the base learning rate 10^4 times the original settings. For example, in DSS the base learning rate is 10^{-8}, while in our F-DSS, the base learning rate is 10^{-4}. All other hyper-parameters are consistent with the original implementations for a fair comparison.

Evaluation metrics. We evaluate the performance of saliency maps in terms of maximal F-measure (MaxF), mean F-measure (MeanF) and mean absolute error (MAE = $\frac{1}{N}\sum_i^N |\hat{y}_i - y_i|$). The factor β^2 in Eq. 1 is set to 0.3 as suggested by [1, 11, 16, 19, 30]. By applying series thresholds $t \in \mathcal{T}$ to the saliency map $\hat{Y}$, we obtain binarized saliency maps $\dot{Y}^t$ with different precisions, recalls and F-measures.

Then the optimal threshold t_o is obtained by exhaustively searching the testing set:

$$t_o = \operatorname*{argmax}_{t \in \mathcal{T}} F(Y, \dot{Y}^t). \tag{12}$$

Finally, we binarize the predictions with t_o and evaluate

	Training data		ECSSD [35]			HKU-IS [15]			PASCALS [17]			SOD [23]			DUT-OMRON [23]		
Model	Train	#Images	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE
Log-FLoss	MB [20]	2.5K	.909	.891	.057	.903	.881	.043	.823	.808	.101	.838	.817	.122	.770	.741	.062
FLoss	MB [20]	2.5K	.914	.903	.050	.908	.896	.038	.829	.818	.091	.843	.838	.111	.777	.755	.067

Table 2. Performance comparison of Log-FLoss (Eq. 8) and FLoss (Eq. 6). FLoss performs better than Log-FLoss on most datasets in terms of MaxF, MeanF and MAE. Specifically FLoss enjoys a large improvement in terms of MeanF because of its high-contrast predictions.

the best F-measure:

$$\text{MaxF} = F(Y, \dot{Y}^{t_o}), \tag{13}$$

where $\dot{Y}^{t_o}$ is a binary saliency map binarized with t_o. The MeanF is the average F-measure under different thresholds:

$$\text{MeanF} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} F(Y, \dot{Y}^{t}), \tag{14}$$

where $\mathcal{T}$ is the collection of possible thresholds.

4.2. Log-FLoss vs FLoss

Firstly we compare FLoss with its alternative, namely Log-FLoss defined in Eq. 8, to justify our choice. As analyzed in Sec. 3.3, FLoss enjoys the advantage of having large gradients in the saturated area that cross-entropy loss and Log-FLoss don't have.

To experimentally verify our assumption that FLoss will produce high-contrast predictions, we train the DSS [11] model with FLoss and Log-FLoss, respectively. The training data is MSRA-B [20] and hyper-parameters are kept unchanged with the original implementation, except for the base learning rate. We adjust the base learning rate to 10^{-4} since our method accept larger learning rate, as explained in Sec. 3.3. Quantitative results are in Table 2 and some example detected saliency maps are shown in Fig. 2.

Although both of Log-FLoss and FLoss use F-measure as maximization target, FLoss derives polarized predictions with high foreground-background contrast, as shown in Fig. 2. The same conclusion can be drawn from Table 2 where FLoss achieves higher Mean F-measure. Which reveals that FLoss achieves higher F-measure score under a wide range of thresholds.

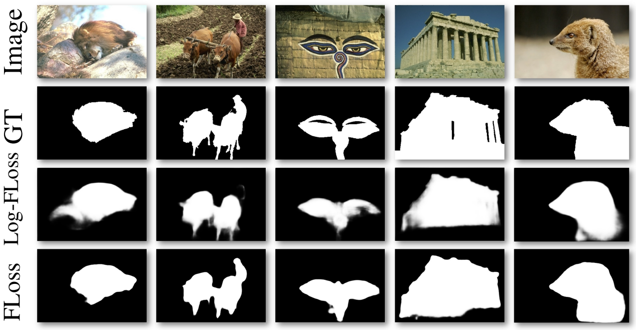


Figure 2. Example saliency maps by FLoss (bottom) and Log-FLoss (middle). Our proposed FLoss method produces high-contrast saliency maps.

4.3. Evaluation results on open Benchmarks

We compare the proposed method with several baselines on 5 popular datasets. Some example detection results are shown in Fig. 3 and comprehensive quantitative comparisons are in Table 3. In general, FLoss-based methods can obtain considerable improvements compared with their cross-entropy loss (CELoss) based counterparts especially in terms of mean F-measure and MAE. This is mainly because our method is stable against the threshold, leading to high-performance saliency maps under a wide threshold range. In our detected saliency maps, the foreground (salient objects) and background are well separated, as shown in Fig. 3 and explained in Sec. 3.3.

4.4. Threshold Free Salient Object Detection

State-of-the-art SOD methods [11, 16, 19, 38] often evaluate maximal F-measure as follows: (a) Obtain the saliency maps $\hat{Y}_i$ with pretrained model; (b) Tune the best threshold t_o by exhaustive search on the testing set (Eq. 12) and binarize the predictions with t_o; (c) Evaluate the maximal F-measure according to Eq. 13.

There is an obvious flaw in the above procedure: the optimal threshold is obtained via an exhaustive search on the testing set. Such procedure is impractical for real-world applications as we would not have annotated testing data. And even if we tuned the optimal threshold on one dataset, it can not be widely applied to other datasets.

We further analyze the sensitivity of methods against thresholds in two aspects: (1) model performance under different thresholds, which reflects the stability of a method against threshold change, (2) the mean and variance of optimal threshold t_o on different datasets, which represent the generalization ability of t_o tuned on one dataset to others.

Fig. 4 (a) illustrates the F-measure w.r.t different thresholds. For most methods without FLoss, the F-measure changes sharply with the threshold, and the maximal F-measure (MaxF) presents only in a narrow threshold span. While FLoss based methods are almost immune from the

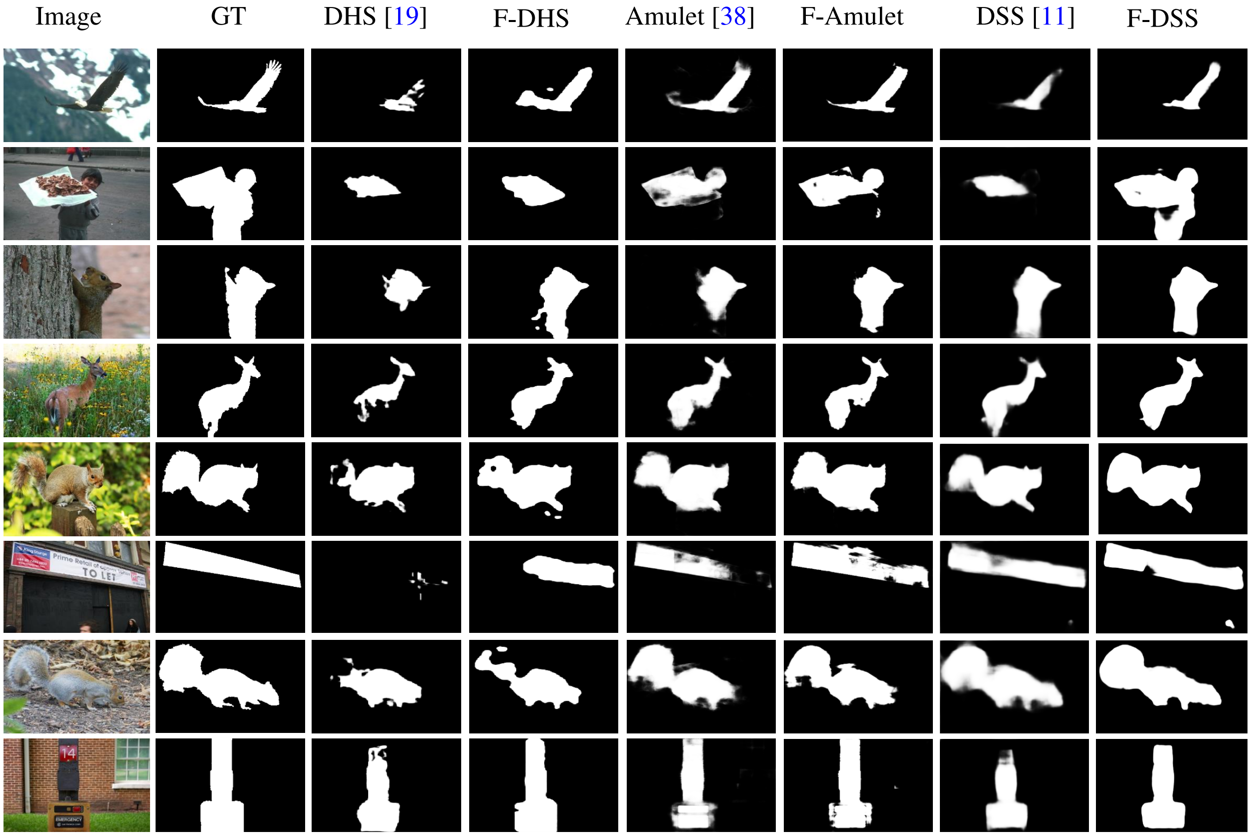


Figure 3. Salient object detection examples on several popular datasets. F-DHS, F-Amulet and F-DSS indicate the original architectures trained with our proposed FLoss. FLoss leads to sharp salient confidence, especially on the object boundaries.

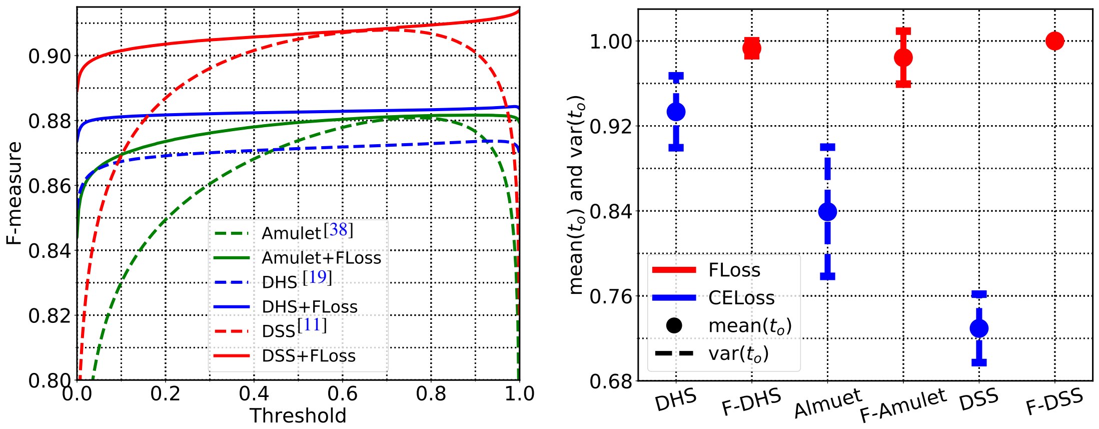
	Training data		ECSSD [35]			HKU-IS [15]			PASCALS [17]			SOD [23]			DUT-OMRON [23]		
Model	Train	#Images	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE
RFCN [30]	MK [5]	10K	.898	.842	.095	.895	.830	.078	.829	.784	.118	.807	.748	.161	-	-	-
DCL [16]	MB [20]	2.5K	.897	.847	.077	.893	.837	.063	.807	.761	.115	.833	.780	.131	.733	.690	.095
DHS [19]	MK [5]+D [23]	9.5K	.905	.876	.066	.891	.860	.059	.820	.794	.101	.819	.793	.136	-	-	-
Amulet [38]	MK [5]	10K	.912	.898	.059	.889	.873	.052	.828	.813	.092	.801	.780	.146	.737	.719	.083
DHS [19]	MB	2.5K	.874	.867	.074	.835	.829	.071	.782	.777	.114	.800	.789	.140	.704	.696	**.078**
DHS+FLoss [19]	MB	2.5K	**.884**	**.879**	**.067**	**.859**	**.854**	**.061**	**.792**	**.786**	**.107**	**.801**	**.795**	**.138**	**.707**	**.701**	.079
Amulet [38]	MB	2.5K	.881	.857	.076	.868	.837	.061	.775	.753	.125	.791	.776	.149	.704	.663	.098
Amulet-FLoss	MB	2.5K	**.894**	**.883**	**.063**	**.880**	**.866**	**.051**	**.791**	**.776**	**.115**	**.805**	**.800**	**.138**	**.729**	**.696**	**.097**
DSS [11]	MB	2.5K	.908	.889	.060	.899	.877	.048	.824	.806	.099	.835	.815	.125	.761	.738	.071
DSS+FLoss	MB	2.5K	**.914**	**.903**	**.050**	**.908**	**.896**	**.038**	**.829**	**.818**	**.091**	**.843**	**.838**	**.111**	**.777**	**.755**	**.067**

Table 3. Quantitative comparison of different methods on 6 popular datasets. Our proposed FLoss consistently improves performance in terms of both MAE (the smaller the better) and F-measure (the larger the better). Especially in terms of Mean F-measure, we outperform the state-of-the-art with very clear margins, because our method is able to produce high-contrast predictions that can achieve high F-measure under a wide range of thresholds.

change of threshold.

Fig. 4 (b) reflects the mean and variance of t_o across different datasets. Conventional methods (DHS, DSS, Amulet) present unstable t_o on different datasets, as evidenced by their large variances. While the t_o of FLoss-based methods (F-DHS, F-Amulet, F-DSS) stay unchanged across different datasets and different backbone network architectures.

In conclusion, the proposed FLoss is stable against



(a) (b)

Figure 4. (a) F-measures under different thresholds on the ECSSD dataset. (b) The mean and variance of optimal threshold t_o. FLoss-based methods hold stable t_o across different datasets (lower t_o variances) and different backbone architectures (F-DHS, F-Amulet and F-DSS hold very close mean t_o).

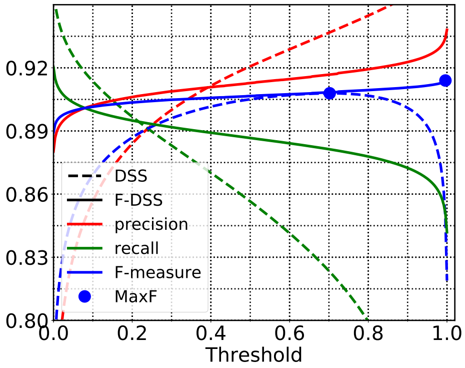


Figure 5. **Precision**, **Recall**, **F-measure** and Maximal F-measure (●) of DSS (- - -) and F-DSS (—) under different thresholds. DSS tends to predict unknown pixels as the majority class–the background, resulting in high precision but low recall. FLoss is able to find a better compromise between precision and recall.

threshold t in three aspects: (1) it achieves high performance under a wide range of threshold; (2) optimal threshold t_o tuned on one dataset can be transferred to others, because t_o varies slightly across different datasets; and (3) t_o obtained from one backbone architecture can be applied to other architectures.

4.5. The Label-unbalancing Problem in SOD

The foreground and background are biased in SOD where most pixels belong to the non-salient regions. The unbalanced training data will lead the model to local minimal that tends to predict unknown pixels as the background. Consequently, the recall will become a bottleneck to the performance during evaluations, as illustrated in Fig. 5.

Although assigning loss weight to the positive/negative samples is a simple way to offset the unbalancing problem, an additional experiment in Table 4 reveals that our method performs better than simply assigning loss weight. We define the *balanced cross-entropy loss* with weight factor between positive/negative samples:

$$\mathcal{L}_{balance} = \sum_{i}^{|Y|} w_1 \cdot y_i \log \hat{y}_i + \\ w_0 \cdot (1 - y_i) \log (1 - \hat{y}_i). \qquad (15)$$

The loss weights for positive/negative samples are determined by the positive/negative proportion in a mini-batch: $w_1 = \frac{1}{|Y|} \sum_{i}^{|Y|} 1(y_i {=}{=} 0)$ and $w_0 = \frac{1}{|Y|} \sum_{i}^{|Y|} 1(y_i {=}{=} 1)$, as suggested in [34] and [28].

4.6. The Compromise Between Precision and Recall

Recall and precision are two conflict metrics. In some applications, we care recall more than precision, while in other tasks precision may be more important than recall. The β^2 in Eq. 1 balances the bias between precision and precision when evaluating the performance of specific tasks. For example, recent studies on edge detection use [2, 34, 28] $\beta^2 = 1$, indicating its equal consideration on precision and recall. While saliency detection [1, 11, 16, 19, 30] usually uses $\beta^2 = 0.3$ to emphasize the precision over the recall.

As an optimization target, the FLoss should also be able to balance the favor between precision and recall. We train models with different β^2 and comprehensively evaluate their performances in terms of precision, recall and F-measure. Results in Fig. 6 reveal that β^2 is a bias adjuster between precision and recall: larger β^2 leads to higher recall while lower β^2 results in higher precision.

4.7. Faster Convergence and Better Performance

In this experiment, we train three state-of-the-art saliency detectors (Amulet [38], DHS [20] and DSS [11])

	Training data		ECSSD [35]			HKU-IS [15]			PASCALS [17]			SOD [23]			DUT-OMRON [23]		
Model	Train	#Images	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE	MaxF	MeanF	MAE
DSS [11]	MB [20]	2.5K	.908	.889	.060	.899	.877	.048	.824	.806	.099	.835	.815	.125	.761	.738	.071
DSS+Balance	MB [20]	2.5K	.910	.890	.059	.900	.877	.048	.827	.807	.097	.837	.816	.124	.765	.741	.069
DSS+FLoss	MB [20]	2.5K	**.914**	**.903**	**.050**	**.908**	**.896**	**.038**	**.829**	**.818**	**.091**	**.843**	**.838**	**.111**	**.777**	**.755**	**.067**

Table 4. Performance comparisons across the original cross-entropy loss (Eq. 10), balanced cross-entropy loss (Eq. 15) and our proposed FLoss (Eq. 6). Original cross-entropy learns a biased prior towards the major class (the background). This is evidenced by the low recall: many positive points are mis-predicted as negative because of biased prior. By assigning loss weights on foreground/background samples, the *balanced cross-entropy loss* can alleviate the unbalancing problem. Our proposed method performs better than the *balanced cross-entropy loss*, because the F-measure criterion can automatically adjust data unbalance.

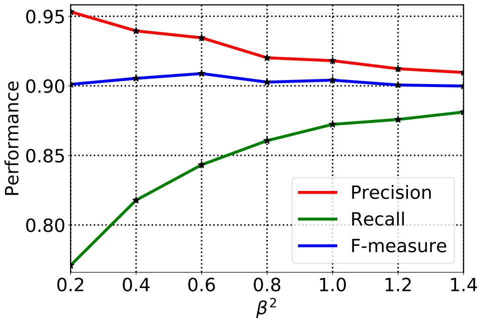


Figure 6. **Precision**, **Recall**, **F-measure** of model trained under different β^2 (Eq. 1). The precision decreases with the growing of β^2 whereas recall increases. This characteristic gives us much flexibility to adjust the balance between recall and precision: use larger β^2 in a recall-first application and lower β^2 otherwise.

and their FLoss counterparts. Then we plot the performance of all the methods at each checkpoint to determine the converge speed and converged performance of respective models. All the models are trained on the MB [20] dataset and tested on the ECSSD [35] dataset. The results are shown in Fig.7.

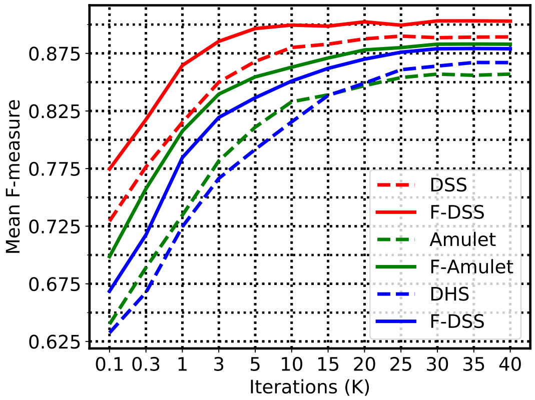


Figure 7. Performance versus training iterations. Our method presents faster convergence and higher converged performance.

We observe that our FLoss offers a per-iteration performance promotion for all the three saliency models. We also find that the FLoss-based methods quickly learn to focus on the salient object area and achieve high F-measure score after hundreds of iterations. While cross-entropy based methods produce blurry outputs and cannot localize salient areas very preciously. As shown in Fig. 7, FLoss based methods converge faster than its cross entropy competitors and get higher converged performance.

5. Conclusion

In this paper, we propose to directly maximize the F-measure for salient object detection. We introduce the FLoss that is differentiable w.r.t the predicted posteriors as the optimization objective of CNNs. The proposed method achieves better performance in terms of better handling biased data distributions. Moreover, our method is stable against the threshold and able to produce high-quality saliency maps under a wide threshold range, showing great potential in real-world applications. By adjusting the β^2 factor, one can easily adjust the compromise between precision and recall, enabling flexibility to deal with various applications. Comprehensive benchmarks on several popular datasets illustrate the advantage of the proposed method.

Future work. We plan to improve the performance and efficiency of the proposed method by using recent backbone models, *e.g.*, [10, 27]. Besides, the FLoss is potentially helpful to other binary dense prediction tasks such as edge detection [21], shadow detection [12] and skeleton detection [40].

Acknowledgment. This research was supported by NSFC (61572264, 61620106008), the national youth talent support program, and Tianjin Natural Science Foundation (17JCJQJC43700, 18ZXZNGX00110).

References

[1] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *CVPR*, pages 1597–1604, 2009.

[2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE PAMI*, 33(5):898–916, 2011.

[3] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, 5(2):117–150, 2019.

[4] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *TIP*, 24(12):5706–5722, 2015.

[5] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE PAMI*, 37(3):569–582, 2015.

[6] Krzysztof Dembczynski, Arkadiusz Jachnik, Wojciech Kotlowski, Willem Waegeman, and Eyke Hüllermeier. Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In *ICML*, pages 1130–1138, 2013.

[7] Krzysztof J Dembczynski, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. An exact algorithm for f-measure maximization. In *NeurIPS*, pages 1404–1412, 2011.

[8] Deng-Ping Fan, Jiangjiang Liu, Shanghua Gao, Qibin Hou, Ali Borji, and Ming-Ming Cheng. Salient objects in clutter: Bringing salient object detection to the foreground. *ECCV*, 2018.

[9] Deng-Ping Fan, Wenguan Wang, Ming-Ming Cheng, and Jianbing Shen. Shifting more attention to video salient object detection. In *CVPR*, pages 8554–8564, 2019.

[10] 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*, 2019.

[11] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip Torr. Deeply supervised salient object detection with short connections. *IEEE TPAMI*, 41(4):815–828, 2019.

[12] Xiaowei Hu, Lei Zhu, Chi-Wing Fu, Jing Qin, and Pheng-Ann Heng. Direction-aware spatial context features for shadow detection. In *CVPR*, June 2018.

[13] Martin Jansche. A maximum expected utility framework for binary sequence labeling. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*, pages 736–743, 2007.

[14] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NeurIPS*, pages 109–117, 2011.

[15] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. *CVPR*, 2015.

[16] Guanbin Li and Yizhou Yu. Deep contrast learning for salient object detection. In *CVPR*, pages 478–487, 2016.

[17] Yin Li, Xiaodi Hou, Christof Koch, James M Rehg, and Alan L Yuille. The secrets of salient object segmentation. In *CVPR*, pages 280–287, 2014.

[18] 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 *CVPR*, 2019.

[19] Nian Liu and Junwei Han. Dhsnet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016.

[20] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *IEEE PAMI*, 33(2):353–367, 2011.

[21] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai, and Jinhui Tang. Richer convolutional features for edge detection. *IEEE TPAMI*, 41(8):1939 – 1946, 2019.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[23] Vida Movahedi and James H Elder. Design and perceptual validation of performance measures for salient object segmentation. In *CVPR-workshop*, pages 49–56, 2010.

[24] James Petterson and Tibério S Caetano. Reverse multi-label learning. In *NeurIPS*, pages 1912–1920, 2010.

[25] James Petterson and Tibério S Caetano. Submodular multi-label learning. In *NeurIPS*, pages 1512–1520, 2011.

[26] José Ramón Quevedo, Oscar Luaces, and Antonio Bahamonde. Multilabel classifiers with a probabilistic thresholding strategy. *Pattern Recognition*, 45(2):876–883, 2012.

[27] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, June 2018.

[28] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *CVPR*, pages 3982–3991, 2015.

[29] Cornelis Joost Van Rijsbergen. Foundation of evaluation. *Journal of Documentation*, 30(4):365–373, 1974.

[30] Linzhao Wang, Lijun Wang, Huchuan Lu, Pingping Zhang, and Xiang Ruan. Saliency detection with recurrent fully convolutional networks. In *ECCV*, pages 825–841, 2016.

[31] Wenguan Wang, Jianbing Shen, Ming-Ming Cheng, and Ling Shao. An iterative and cooperative top-down and bottom-up inference network for salient object detection. In *CVPR*, pages 5968–5977, 2019.

[32] Wenguan Wang, Jianbing Shen, Xingping Dong, and Ali Borji. Salient object detection driven by fixation prediction. In *CVPR*, pages 1711–1720, 2018.

[33] Wenguan Wang, Shuyang Zhao, Jianbing Shen, Steven CH Hoi, and Ali Borji. Salient object detection with pyramid attention and salient edges. In *CVPR*, pages 1448–1457, 2019.

[34] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *ICCV*, pages 1395–1403, 2015.

[35] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013.

[36] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[37] Nan Ye, Kian Ming A Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing f-measures: a tale of two approaches. In *ICML*, pages 1555–1562, 2012.

[38] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet: Aggregating multi-level convolutional features for salient object detection. *ICCV*, 2017.

[39] Jiaxing Zhao, Jiangjiang liu, Dengping Fan, Yang Cao, Jufeng Yang, and Ming-Ming Cheng. Egnet:edge guidance network for salient object detection. In *ICCV*, Oct 2019.

[40] Kai Zhao, Wei Shen, Shanghua Gao, Dandan Li, and Ming-Ming Cheng. Hi-Fi: Hierarchical feature integration for skeleton detection. In *IJCAI*, pages 1191–1197, 7 2018.