Uncertainty-aware Joint Salient Object and Camouflaged Object Detection

Aixuan Li¹,‡ Jing Zhang²,³,‡ Yunqiu Lv¹ Bowen Liu¹ Tong Zhang⁴ Yuchao Dai¹✉
¹ Northwestern Polytechnical University, China ² Australian National University, Australia ³ CSIRO, Australia ⁴ EPFL, Switzerland
‡ Equal contributions; ✉ Corresponding author: daiyuchao@nwpu.edu.cn

Abstract

Visual salient object detection (SOD) aims at finding the salient object(s) that attract human attention, while camouflaged object detection (COD) on the contrary intends to discover the camouflaged object(s) that hidden in the surrounding. In this paper, we propose a paradigm of leveraging the contradictory information to enhance the detection ability of both salient object detection and camouflaged object detection. We start by exploiting the easy positive samples in the COD dataset to serve as hard positive samples in the SOD task to improve the robustness of the SOD model. Then, we introduce a “similarity measure” module to explicitly model the contradicting attributes of these two tasks. Furthermore, considering the uncertainty of labeling in both tasks’ datasets, we propose an adversarial learning network to achieve both higher order similarity measure and network confidence estimation. Experimental results on benchmark datasets demonstrate that our solution leads to state-of-the-art (SOTA) performance for both tasks¹.

1. Introduction

Visual salient object detection (SOD) aims to localize the most salient region(s) of the images that attract human attention. To be qualified as a “salient” object, one should have high contrast compared with its global and local context. The camouflaged objects oppositely usually share similar structure or texture information with the environment, which try hard to fade themselves into the local context. In this way, the SOD models [48, 46, 42, 38] are designed based on both global contrast and local contrast, while the COD models [11, 26, 35, 52, 34] usually avoid searching the camouflaged objects in those salient regions. We notice that a higher level of saliency indicates a lower level of camouflage and vice versa as shown in Fig. 1. This observation shows that the salient objects and camouflaged objects are two contradicting categories of objects. However, there still exist objects that are both salient and camouflaged, e.g. the polar bear in the middle of Fig. 1, which indicates that these two tasks are partially positively related at the dataset level.

Existing SOD models [48, 47, 59, 46, 38] mainly focus on two directions: 1) building effective saliency network [47, 59] for accurate saliency detection with pixel-wise accuracy constraint; and 2) designing appropriate loss functions [46, 38] to achieve structure-preserving saliency detection. The former digs into network structure, while the latter cares more about network loss function. We argue that an effective training dataset can lead to more performance gain in addition to network structure design or loss function, it’s the training data that is regressed.

One typical solution to explore the training dataset is data augmentation, which usually involves linear or non-linear transformation of the dataset. We find that, although performance improvement can be obtained with some basic data augmentation, e.g. image flipping, rotation, cropping, and etc., none of these methods are specially designed for saliency detection. As a context-based task, a more effective data augmentation technique should be context-aware. For SOD, the salient objects are those that can be easily detected, or the high-contrast objects as shown in Fig. 1. We intend to augment the dataset to include lower contrast samples. Considering the partial positively related attribute of SOD and COD at dataset level, we intend to design a joint learning framework to learn both tasks and select easy

¹Our code is publicly available at: https://github.com/JingZhang617/Joint_COD_SOD

Figure 1. Illustration of the transition from camouflaged objects to salient objects, where the image in the middle could belong to both camouflaged object dataset and salient object dataset.
samples from COD (e.g. the polar bear) as hard samples for SOD, achieving contrast-level data augmentation.

Joint training is mostly designed for positively related tasks [43, 14, 51]. In contrary, we propose to integrate two contradicting tasks (SOD and COD) into one network with a “Similarity measure” module as shown in Fig. 2. The basic assumption of our similarity measure module is that the activated regions of the same image for the two tasks should be different, leading to latent features apart from each other. To this end, we introduce the third dataset, e.g. PASCAL VOC 2007 images [7] in particular, to our framework serving as the connection modeling dataset. The goal of these extra images is to achieve similarity measures and force the two tasks to focus on different regions of the image.

Moreover, as shown in Fig. 1, the salient object is salient in both local and global contexts, while the camouflaged object is hiding in its local context. Due to the high contrast, the local context of the salient objects is easier to model than that of the camouflaged objects, as there exists no clear boundary between camouflaged objects and their surroundings. By jointly training a salient object detection network and camouflaged object detection network, the salient object branch can learn precise local context information for accurate camouflaged object detection.

Lastly, we observe two types of uncertainty while labeling the dataset for each task. For salient object detection, the subjective nature of saliency [55, 54, 53] leads to ambiguity of prediction, as shown in Fig. 5(a). For camouflaged object detection, the uncertainty comes from the difficulty in fully annotating the camouflaged objects as they usually share similar color or texture with the environment, as shown in Fig. 5(c). We then introduce adversarial training to explicitly model the confidence of network predictions, and estimate model uncertainty for both tasks.

We summarize our main contributions as: 1) We introduce the first joint salient object detection and camouflaged object detection network within an adversarial learning framework to explicitly model prediction uncertainty of each task. 2) We design the similarity measure module to explicitly model the “contradicting” attributes of the two tasks. 3) We present a data interaction strategy and treat easy samples from camouflage dataset as hard samples for saliency detection, achieving a robust saliency model.

2. Related Work

Salient Object Detection Models Existing deep saliency detection models [47, 48, 42, 29, 46, 12, 38] are mainly designed to achieve structure-preserving saliency predictions. [38, 48] introduced auxiliary edge detection branch to produce a saliency map with precise structure information. Wei et al. [46] presented structure-aware loss function to penalize prediction along object edges. Wu et al. [47] designed a cascade partial decoder to achieve accurate saliency detection with finer detailed information. Feng et al. [12] proposed a boundary-aware mechanism to improve the accuracy of network prediction on the boundary. There also exist salient object detection models that benefit from data of other sources. [45, 24] integrated fixation prediction and salient object detection in a unified framework to explore the connections of the two related tasks. Zeng et al. [51] presented to jointly learn a weakly supervised semantic segmentation and fully supervised salient object detection model to benefit from both tasks.

Camouflaged Object Detection Models Camouflage models are designed to discover the camouflaged object(s) hidden in the surrounding. Different from salient objects, which are those attracting human attention, camouflaged objects are those trying to decrease the conspicuousness. The concept of camouflage is usually associated with context [1, 2, 5]. Cuthill et al. [6] presented that an effective camouflage includes two mechanisms: the background pattern matching, where the color is similar to the environment, and the disruptive coloration, which usually involves bright colors along edge, and makes the boundary between camouflaged objects and the background unnoticeable. Bhajantri et al. [3] utilized co-occurrence matrix to detect defective. Pike et al. [37] combined several salient visual features to quantify camouflage, which could simulate the visual mechanism of a predator. In the field of deep learning, Fan et al. [11] proposed the first publicly available camouflage deep network with the largest camouflaged object training set.

Multi-task Learning The basic assumption behind multi-task learning is that there exists shared information among different tasks. In this way, multi-task learning is widely used to extract complementary information about positively related tasks. Kalogeiton et al. [22] jointly detected objects and actions in a video scene. Zhen et al. [58] designed a joint semantic segmentation and boundary detection framework by iteratively fusing feature maps generated for each task with pyramid context module. In order to solve the problem of insufficient supervision in semantic alignment and object landmark detection, Jeon et al. [19] designed a joint loss function to impose constraints between tasks, and only reliable matched pairs were used to improve the model robustness with weak supervision. Joung et al. [21] solved the problem of object viewpoint changes in 3D object detection and viewpoint estimation with a cylindrical convolutional network, which obtains view-specific features with structural information at each viewpoint for both two tasks. Luo et al. [32] presented a multitask framework for referring expression comprehension and segmentation.

Adversarial Learning Adversarial learning is an effective solution to improve the robustness of the deep neural network. In fully supervised models, adversarial learning can measure higher-order inconsistencies between labels and predictions [25]. Specifically, Li et al. [28] introduced
to learn SOD and COD in a uni-
work as shown in Fig. 3. Our Method
dicting attributes with a similarity measure module.
joint learning framework by explicitly modeling the contra-
data for semi-supervised learning.
regions of semantic segmentation predictions of unlabeled
from the fake saliency map (prediction). Similarly, Jiang
was used to distinguish the real saliency map (ground truth)
detection network, where the fully connected discriminator
for salient object detection. 

3.2. Contradicting modeling

3.1. Data interaction as data augmentation

As shown in Fig. 1, there exist samples in the COD
dataset that are both salient and camouflaged. We argue
that those samples can be treated as hard samples for salient
object detection to achieve robust learning. To select those
samples from the COD dataset, we resort to Mean Absolute
Error (MAE), and select samples in COD dataset [11] which
achieve the smallest MAE by testing it using a trained SOD
model [48]. Specifically, for camouflaged object detection
training dataset \(D_c = \{X^c_i, Y^c_i\}_{i=1}^{N_c}\), where \(i\) indexes the
images, and \(N_c\) is the size of camouflaged object detection
training set. We defined the trained SOD model as \(M_{\theta_s}\).
Then we obtain saliency prediction of the images in \(D_c\) as
\(P^c_s = M_{\theta_s}(X^c) = \{p^c_s_i\}_{i=1}^{N_c}\), where \(p^c_s\) is the saliency pre-
diction in COD training dataset. We assume that easy sam-
pies for COD can be treated as hard samples for SOD as
shown in Fig. 1. Then we select \(M = 400\) samples \(D^M_c\) of
the smallest MAE in \(D_c\), and replace it with \(M = 400\) ran-
domly selected samples in our SOD training dataset [44] as
a data augmentation technique. We show the selected sam-
pples in Fig. 3, which clearly illustrates the partially positive
connection of the two tasks at the dataset level.

3.2. Contradicting modeling

Similar as above, let’s define our camouflaged object de-
tection training dataset as \(D_c = \{X^c_i, Y^c_i\}_{i=1}^{N_c}\) and aug-
mented salient object detection training dataset as \(D_s =
\{X^s_i, Y^s_i\}_{i=1}^{N_s}\), where \(i\) indexes images. \(X\) and \(Y\) are the im-
generic noise to destroy adversarial perturbation of the input
for salient object detection. [31, 41] built the Generative Adversarial
Network (GAN)[16] based saliency detection network, where the fully connected discriminator
was used to distinguish the real saliency map (ground truth)
from the fake saliency map (prediction). Similarly, Jiang
et al. [20] designed a GAN based framework for RGB-
D saliency detection to solve the cross-modality detection
problem. In the case of insufficient annotations, adversarial
learning can serve as guidance to select confident samples or
generate new samples. Souly et al. [40] used adversarial
learning to generate fake images with image-level labels and
noise, which in turn can make the real samples gathering
in feature space and improve accuracy for weakly super-
vised semantic segmentation. [18, 36] treated adversarial
learning as a confidence measure to obtain the trustworthy
regions of semantic segmentation predictions of unlabeled
for semi-supervised learning.

Different from existing multi-task learning frameworks
that mainly benefit from positively related tasks, we instead
build the connections of two contradicting tasks within a
joint learning framework by explicitly modeling the con-
trdicting attributes with a similarity measure module.

3. Our Method

We design an uncertainty-aware joint learning frame-
work as shown in Fig. 2 to learn SOD and COD in a uni-
fied framework. Firstly, as a data augmentation technique,
we select a group of easy samples from the COD training
data set to achieve robust SOD. Then, we present the “Simi-
ilarity measure” module to explicitly model the “contradict-
ing” attributes of the two tasks. Lastly, we introduce our
uncertainty-aware adversarial training network to produce
interpretable predictions during testing, and achieve higher-
order similarity measure during training.
Figure 4. Detected region from the saliency encoder branch (first row) and the camouflage encoder branch (second row).

age and ground truth pair, \(N_c\) and \(N_s\) are size of the camouflage training set and saliency training set respectively. Based on the “contradicting” attribute of SOD and COD, we design a “Similarity measure” module in Fig. 2 to explicitly model the connection of the two tasks.

Specifically, we introduce another set of images from PASCAL VOC 2007 dataset [7] as “connection modeling” dataset \(D_p\) = \{\(X'^p\)\}_{i=1}^N, from which we extract the camouflaged feature and the salient feature. With the three datasets (COD dataset \(D_c\), augmented SOD dataset \(D_s\) and connection modeling dataset \(D_p\)), our contradicting modeling framework uses the “Feature encoder” module to extract both camouflage feature and saliency feature, and then use the “Similarity measure” module to model connection of the two tasks with the connection modeling dataset.

**Feature Encoder**

We design both the saliency encoder \(E_{\alpha_s}\) and camouflage encoder \(E_{\alpha_c}\) network with the same backbone network, e.g. the ResNet50 [17], where \(\alpha_s\) and \(\alpha_c\) are network parameter sets of each of them respectively. Initially, the ResNet50 backbone network has four groups of convolutional layers of channel size 256, 512, 1024 and 2048 respectively. We then define the output features of both encoders as \(F_{\alpha_s} = \{f_1^s, f_2^s, f_3^s, f_4^s\}\) and \(F_{\alpha_c} = \{f_1^c, f_2^c, f_3^c, f_4^c\}\), where \(f_k, k = 1, ..., 4\) is feature map of the \(k\)-th group.

**Similarity Measure**

Different from the feature encoder module, which takes images from \(D_c\) and \(D_s\) as input to produce task-specific feature maps, the similarity measure module \(S_\theta\) takes the connection modeling data \(D_p\) as input to model the connection of SOD and COD, where \(\theta\) is parameter set of the similarity measure module. Given the trained saliency encoder \(E_{\alpha_s}\) and camouflage encoder \(E_{\alpha_c}\), the saliency feature and camouflage feature of images \(X^p\) are \(F^p_{\alpha_s} = \{f_{p_1}^s, f_{p_2}^s, f_{p_3}^s, f_{p_4}^s\}\) and \(F^p_{\alpha_c} = \{f_{p_1}^c, f_{p_2}^c, f_{p_3}^c, f_{p_4}^c\}\) respectively. We then concatenate each of above two features channel-wise and feed them to the same fully connected layer to obtain the latent saliency feature and latent camouflage feature of \(X^p\) as \(f^{sp} = S_\theta(F^p_{\alpha_s})\) and \(f^{cp} = S_\theta(F^p_{\alpha_c})\) respectively. Empirically, we set the dimension of the latent space as \(K = 700\). For the same image in \(D_p\), we assume that the SOD network and COD network should focus on different regions, leading to different feature representation. Then, we choose the cosine similarity to measure the difference between the saliency feature and the camouflage feature in latent space, and define the latent space loss as:

\[
\mathcal{L}_{latent} = \cos(f^{sp}, f^{cp}) = \frac{f^{sp} \cdot f^{cp}}{\|f^{sp}\| \times \|f^{cp}\|}.
\]

In Fig. 4, we show the activation region (the processed predictions) of the same image from both the saliency encoder (first row) and camouflage encoder (second row). Specifically, given same image \(X^p\), we compute it’s camouflage map and saliency map, and highlight the detected foreground region in red. Fig. 4 shows that the two encoders focus on different regions of the image, where the saliency encoder pays more attention to the region that stand out from the context, and the camouflage encoder focuses more on the hidden object with similar color or structure as the background, which is consistent with our assumption that these two tasks are contradicting with each other in general.

**3.3. Uncertainty-aware adversarial learning**

As shown in Fig. 5, for the SOD dataset, the uncertainty comes from the ambiguity of saliency ((a) (b)), and for the COD dataset, the uncertainty results from the difficulty of annotation ((c) (d)). e.g. the ball in the orange rectangle (a) can be defined as salient, but it’s background in (b). The orange region in (c) belongs to the camouflaged object, while it’s too similar to the background, making it very difficult to create the accurate annotation. We then introduce an uncertainty-aware adversarial training strategy to model the task-specific uncertainty in our joint learning framework, which includes a “Prediction decoder” module to produce task-related predictions, a “Confidence estimation” module to estimate uncertainty of each prediction, and an adversarial learning strategy for robust model training.

**Prediction decoder**

As shown in Fig. 2, we design a shared decoder structure for joint SOD and COD learning. We argue that the different “Feature encoder” modules can generate task-specific features for COD images and SOD images. Then the “Prediction decoder” module aims to integrate the task-specific feature with their corresponding lower level feature to produce predictions. Specifically, given task-specific feature
For the prediction decoder module, we first have the task-specific loss function to learn each task. Specifically, we adopt the structure-aware loss function [46] for both SOD and COD, and define the loss function as:

\[
\mathcal{L}_{str}(Pred, Y) = \omega \cdot \mathcal{L}_{ce}(Pred, Y) + \mathcal{L}_{iou}(Pred, Y),
\]

where \(\omega\) is the edge-aware weight, which is defined as \(\omega = 1 + 5 \cdot \left(\frac{avg\_pool(Y) - Y}{1}\right)\), \(\mathcal{L}_{ce}\) is the cross-entropy loss, \(\mathcal{L}_{iou}\) is the boundary-IOU loss [33], which is defined as:

\[
\mathcal{L}_{iou} = 1 - \frac{\omega \cdot \text{inter} + 1}{\omega \cdot \text{union} - \omega \cdot \text{inter} + 1}.
\]

To achieve adversarial learning, we have adversarial loss for both SOD and COD, which is defined as cross-entropy loss between network prediction and the pre-defined “real” indicator as:

\[
\mathcal{L}_{\alpha}^{adv} = \mathcal{L}_{ce}(D_{\gamma}^{f}(G_{\beta}(F_{\alpha})), 1),
\]

\[
\mathcal{L}_{\beta}^{adv} = \mathcal{L}_{ce}(D_{\gamma}^{f}(G_{\beta}(F_{\alpha})), 1),
\]

respectively for each task, where 1 is an all-one matrix. In this way, the discriminator takes model prediction as input, and tries to recognize it as real ground truth.

For the confidence estimation module, similar to the typical definition of discriminator in GAN [16], we want it to clearly distinguish model prediction and ground truth map. Then, the adversarial loss for the confidence estimation module of the SOD task is defined as:

\[
\mathcal{L}_{\alpha}^{dis} = \mathcal{L}_{ce}(D_{\gamma}^{f}(G_{\beta}(F_{\alpha})), 0) + \mathcal{L}_{ce}(D_{\gamma}^{f}(Y^{s}), 1).
\]

We then have the adversarial loss for the COD task as:

\[
\mathcal{L}_{\beta}^{dis} = \mathcal{L}_{ce}(D_{\gamma}^{f}(G_{\beta}(F_{\alpha})), 0) + \mathcal{L}_{ce}(D_{\gamma}^{f}(Y^{c}), 1).
\]

### Objective Function

As shown in Fig. 2, the two tasks have separate encoder, shared decoder as well as the shared confidence estimation.

### Table 1. Network structure of the discriminator network.

<table>
<thead>
<tr>
<th>Input Channel</th>
<th>Output Channel</th>
<th>kernel size</th>
<th>Stride</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>64</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Algorithm 1 Uncertainty-aware joint salient object detection and camouflaged object detection

**Input:** Training images set: \( D_s = \{X^s, Y^s\} \), \( D_c = \{X^c, Y^c\} \) and \( D_p = \{X^p\} \); Maximal number of learning iterations \( T \).

**Output:** \( \alpha_s, \alpha_c \) for feature encoder, \( \theta \) for similarity measure, \( \beta \) for prediction decoder, and \( \gamma \) for confidence estimation;

1. Initialize \( \alpha_s, \alpha_c \) with ResNet50 [17], and \( \theta, \beta, \gamma \) randomly.
2. for \( t \leftarrow 1 \) to \( T \) do
3. Compute latent loss in Eq. (1) and update \( \alpha_s, \alpha_c, \theta \)
4. Compute generator loss and adversarial loss for SOD according to Eq. (15), and update \( \alpha_s, \beta \)
5. Compute adversarial loss for confidence estimation module according to Eq. (13) and update \( \gamma \)
6. Compute generator loss and adversarial loss for COD according to Eq. (16), and update \( \alpha_c, \beta \)
7. Compute adversarial loss for confidence estimation module according to Eq. (14) and update \( \gamma \)
8. end for

4. Experimental Results

4.1. Setting:

**Dataset:** For salient object detection, we train our model using the augmented DUTS training dataset [43], and testing on six other testing dataset, including the DUTS testing dataset, ECSSD [49], DUT [50], HKU-IS [27], THUR [4] and SOC testing dataset [8]. For camouflaged object detection, we train our model using COD10K training set [11], and test on three camouflaged object detection testing sets, including CAMO [26], CHAMELEON [39], and COD10K.

**Evaluation Metrics:** We use four evaluation metrics to evaluate the performance of both SOD and COD models, including Mean Absolute Error, Mean F-measure, Mean E-measure [10] and S-measure [9].

**Training details:** We train our model in Pytorch with ResNet50 [17] as backbone as shown in Fig. 2. Both the encoders for saliency and camouflage branches are initialized with ResNet50 [17] trained on ImageNet, and other newly added layers are randomly initialized. We resize all the images and ground truth to \( 352 \times 352 \). The maximum iteration is 36000, and we iteratively update 3 times the saliency branch and then one time the camouflage branch. The initial learning rate is 2.5e-5. We adopt the “step” learning rate decay policy, and set the decay step as 24000 iteration, and decay rate as 0.1. The whole training takes 8 hours with batch size 15 on an NVIDIA GeForce RTX 2080 GPU.

4.2. SOD performance comparison

We compare performance of our SOD branch with eleven SOTA SOD models as shown in Table 2. One observation from Table 2 is that the structure-preserving strategy is widely used in the state-of-the-art saliency detection models, e.g. SCRN [48], F3Net [46], ITSD [59], and it can indeed improve model performance. Table 2 shows that we achieve 5/6 best performance, except on SOC testing dataset [8]. The main reason is that there exists texture images in SOC, which may be treated as camouflaged object, thus influence our performance. We will investigate this issue further. Further, we show predictions of ours and SOTA models in Fig. 6, where the “Uncertainty” is obtained based on the prediction from the discriminator. Specifically, we define magnitude of gradient of the discriminator output as uncertainty following [23]. Fig. 6 shows that we produce both accurate prediction and reasonable uncertainty estimation, where the brighter area of the uncertainty map indicates the less confident region.

4.3. COD performance comparison

As there exists only one open source deep camouflaged object detection network (SINet [11] in particular), we retrain existing saliency detection models with the camouflaged object detection training dataset [11], and test on the existing camouflaged object detection testing set. The performance of these models is shown in Table 3 and Fig. 7. The consistent best performance of our camouflage model further illustrates effectiveness of the joint learning framework. Moreover, the produced uncertainty map clearly represents model confidence towards the current prediction, leading to interpretable prediction for the downstream tasks.
4.4. Ablation study

We present extra experiments to fully explore our model, and show performance in Table 4 and Table 5.

**Train each task separately:** We use the same “Feature encoder” and “Prediction decoder” in Fig. 2 to train the SOD network and the COD network separately, and show their performance as “ASOD” and “SCOD” respectively. We also train the saliency model using the original DUTS training dataset [44], and show its performance as “SSOD”. We notice a consistent performance improvement of “ASOD” compared with “SSOD”, which indicates the effectiveness of our contrast-level data augmentation technique. The comparable performance of “ASOD” and “SCOD” with their corresponding SOTA models further prove superior performance of our new network structure.

**Joint training of SOD and COD:** We train the “Feature encoder” and “Prediction decoder” within a joint learning pipeline. The performance is shown as “JSOD1” and “JCOD1” for the saliency detection task and camouflaged object detection task respectively. The improved performance of “JSOD3” and “JCOD3” compared with “ASOD” and “SCOD” indicates that the joint learning framework can further boost performance of each task.

**Joint training of SOD and COD with similarity measure:** We add the task connection constraint to the joint learning framework, e.g. the similarity measure module in particular, and show performance as “JSOD2” and “JCOD2” respectively. In general, we can observe improved performance, especially for the COD10K dataset [11], which verifies effectiveness of our similarity measure module.

**Uncertainty-aware joint training of SOD and COD:** Based on the joint learning framework, we include the adversarial learning module, and show performance as “JSOD3” and “JCOD3”. We observe relative comparable performance with the adversarial learning framework. This mainly lies in the difficulty in training the
adversarial learning branch. We provide the fully convolutional discriminator in Table 1, and set loss for the adversarial learning empirically. A better solution could be searching for a more effective discriminator and weight, which will be our future direction.

### 4.5. Hyper-parameters analysis

In our joint learning framework, we have several hyper-parameters that influence our performance, including the maximum iteration, the interval to iteratively train the saliency and camouflage branch, the base learning rate, the dimension of the latent space, the weight of both adversarial loss and latent loss. We found that due to the different dataset sizes and convergence rates, the maximum iteration has a great impact on the COD task. The SOD training dataset contains 10,553 images, which is 2.5 times of the dataset. Lastly, we presented an adversarial learning network to explicitly model the confidence of network predictions to address the uncertainty in SOD and COD annotations. Experimental results on six benchmark SOD datasets and three benchmark COD datasets demonstrate the effectiveness of our joint learning solution.

### 5. Conclusion

In this paper, we have proposed the first joint salient object detection and camouflaged object detection network within an uncertainty-aware framework. First, we showed that the easy samples in COD dataset could be used as hard samples for SOD to learn robust SOD model. Second, by considering the contradicting attributes of these two tasks, we presented a similarity measure module to explicitly build the task connection with the extra connection modeling dataset. Lastly, we presented an adversarial learning network to explicitly model the confidence of network predictions to address the uncertainty in SOD and COD annotations. Experimental results on six benchmark SOD datasets and three benchmark COD datasets demonstrate the effectiveness of our joint learning solution.

### Acknowledgements

This research was supported in part by National Natural Science Foundation of China (61871325), National Key Research and Development Program of China (2018AAA0102803), CSIRO’s Machine Learning and Artificial Intelligence Future Science Platform (MLAI FSP), and the Swiss National Science Foundation via the Sinergia grant CRSII5-180359. We would like to thank the anonymous reviewers for their useful feedbacks.
References


[38] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019. 1, 2, 7


[48] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, October 2019. 1, 2, 3, 6, 7, 8


