Weakly Supervised Camouflaged Object Detection Via Bayesian Network

A thesis submitted in part fulfilment of the degree of
Master of Machine Learning and Computer Vision

by
Tao Yu
U7044148

Supervisor: Prof. MiaoMiao Liu, Jing Zhang
Examiner: Prof. MiaoMiao Liu, Jing Zhang

Australian National University

College of Engineering and Computer Science
The Australian National University

November 2021

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of the author's knowledge, it contains no material previously published or written by another person, except where due reference is made in the text.

Tao Yu
13 November 2021

Acknowledgements

I would first like to thank my thesis advisor Dr. Jing Zhang at Australian National University.

From the topic selection to the final completion of the thesis, she gave me dedicated guidance. Her rigorous academic attitude has profoundly affected me and will definitely have an impact on my future study, work and life.

I would also like to acknowledge Dr. Miaomiao Liu of the College of Engineering and Computer Science at Australian National University as the second reader of this thesis. Miaomiao Liu is a respected professor and I am happy to have such an awesome examiner on my thesis.

Finally, I must express my thanks to my parents, my friend Han Zhang and my girlfriend Dr. Chen Chen for providing me with supports and encouragement during my study period. They teach me never to give up. I can not finish my thesis and studies without their help. Thanks so much.

Author
Tao Yu

Abstract

Object detection is a very classical task in the computer vision aspect which is also widely used in people's daily life. However, due to the cost of the ground-truth label, object detection tasks may consume researchers a lot of money, which makes weakly supervised Learning becomes more and more popular. To reduce the burden brought from the labels, we present a weakly supervised camouflaged object detection model via some relatively cheap labels like bounding boxes and scribbles. By combining these two kinds of labels we finally generate a new and strong label that is suitable for camouflaged object detection. Because the difference between the pseudo labels and ground truth is irreparable, people can not make sure how much they should trust in their model's results. With the help of the Bayesian network, we measure the uncertainty caused by the data itself and model by using Monte Carlo Drop out. In addition, we also propose some loss functions to help the model learn more about the structure and recover the original ground truth by handling the uncertainty. Finally the comparative experiment results show the enhancement that was brought from our model.

Contents

List of Figures

List of Tables

Nomenclature

NN	neural network
CNN	convolutional neural network
BNN	bayesian neural network
MB+	fast Minimum Barrier Distance Transform algorithm
MIL	multiple instance learning
CCNN	Constrained convolutional neural network
SINet	Search and Identification Net)
GMM	gaussian mixture model

Introduction

Camouflage is a way for the animal to hide or trick other animals. No wonder for a predator or prey, camouflage is a necessary ability for them to survive. There are 3 main kinds of camouflage in nature including crypsis, aposematism and mimicry. Crypsis is an ability that animal's skin colour will change with the environment like the Arctic hare turns white in winter to adapt to the snowy living environment in the cold winter. Aposematism is a way to make animals themselves more salient, bright colours on their skin can be a warning signal on predators to show they are toxic or dangerous. Mimicry is to mimic some other specific features, to confuse predator or prey from recognizing them. To make better use of camouflage, camouflaged object detection is needed for some cases to detect the camouflaged objects with crypsis and mimicry ability. Because the camouflaged object has a high degree of similarity with the background, the recognition and detection of the camouflaged target are more difficult than our traditional target detection, but it is also more challenging. Examples are shown in Fig. 1.1

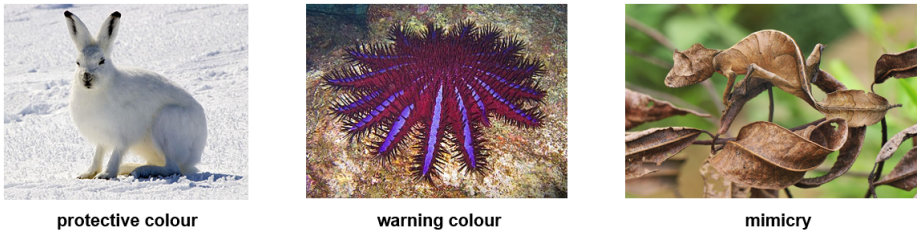

Figure 1.1: Different kinds of camouflage

Nowadays, The main idea to construct a camouflaged object detection network is to use fully-supervised learning. It is true that directly using pixel-wise ground truth can really help the network to learn more about images features, but relying too much on pixel-wise ground truth is not sensible due to the heavy cost. For example, recently, a very large number of labelled images are required during the training process, that is, the training images are generally required to have accurate ground truth. Take the image segmentation task as an example, it is very difficult to obtain a large number of fully labelled images. On the ImageNet dataset, there are 14 million images with category labels, and 500,000 images with bounding boxes, but only 4460 images There are pixel-level segmentation results that can prove that it is very time-consuming to label each pixel in the training image Chum et al. (2019). This problem becomes more serious on camouflaged object detection, that the difficulty of the labelling process has been increased by the high similarity between the foreground and background on camouflaged images.

To relieve stress from annotations, weakly supervised learning is proposed which allow the model to learn some preliminary labelled datasets. In this case, the labelling process of training data is

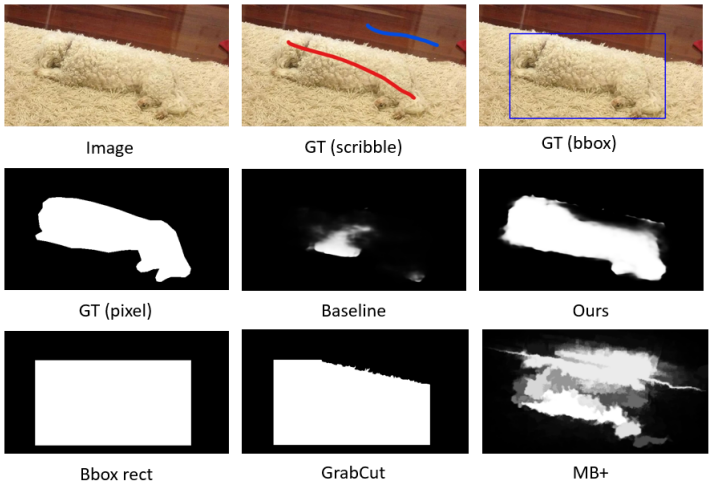

Figure 1.2: Different kinds of annotations (scribble, bounding box, pixel-wise ground truth and bounding box rectangle), unsupervised learning segmentation results (GrabCut, MB+) and supervised learning segmentation result (Baseline, Ours).

very simple, which can greatly reduce the time spent on training data labelling. Common weakly supervised segmentation labels like the image-level label, which gives objects that are contained in an image; bounding box label, which gives the bounding box contains an object; scribbles label, which marks some pixels of the object area in the image, such as drawing some lines, graffiti, etc. Related images are shown in the first row of Fig. 1.2

However, no matter how close are people's generated pseudo labels with the original ground truth, there must be some errors caused by the image label itself since people is hard to generate a real true label without any noise. This problem is obviously exposed in the detection of camouflaged objects since many good unsupervised image segmentation methods work well on common objects but not on camouflaged objects. Most of the weakly-supervised learning methods focus on how to generate a good pseudo label but ignore the importance of the data uncertainty. A model can be trained by pseudo labels but people can not make sure how much they should trust their model because our traditional deep learning algorithms can only give a specific result without how confident the result is. What is more, suppose people can generate good labels with is same as the original ground truth, it is still impossible for them to predict results without loss. This problem comes from the model itself which most of the networks is trying to solve. A common method is to use BNN (Bayesian Neural Network), which makes the weight of each parameter in the network will no longer be a specific number, but will be replaced by a prior distribution. In this way, the network people train will no longer be a function, but a distribution of functions and then we can get a confidence of the predictions. Related images are shown in the second row and last row of Fig. 1.2

In this paper, we propose a method to generate pseudo labels by using bounding box labels and scribble labels. To further improve the label's accuracy, we introduce MB+ method Zhang et al. (2015) to provide more supplementary information and then use DenseCRF Philipp and Koltun (2012) to modify our labels. To quantify the uncertainty caused by the data itself, we create an extra decoder to output uncertainty directly. Monte Carlo Dropout is used to perform Bayesian inference, and we also apply appropriate loss function to help our model to handle the uncertainty and reduce loss. The final results show the advanced improvement which was brought by our network.

In summary, here are our main contributions:

1) We proposal a new pseudo labels' generation method in camouflaged object detection based on GrabCut Rother et al. (2004).

2) We refer the data uncertainty concept in camouflaged object detection and use pseudo label to approximate predict the true distribution of camouflaged objects.

3) We proposal a new combination of loss function to train camouflaged object detection model.

Literature Review

This section will provide some related work on the camouflaged object detection Model and weakly-supervised learning on image segmentation.

2.1 Weakly Supervised Learning on Image Segmentation

To get rid of dependence from annotations, weakly supervised learning can be very good access to solve this problem. In weakly-supervised learning, some of the labels are using bounding boxes, some use scribbles and some use Image-level labels. Hsu et al. (2019) proposal an instance segmentation based on bounding box leveraged multiple instance learning (MIL) that formulated the use of bounding box to predict objects and use the tightness of bounding box to develop the MIL formula and integrate it into instance segmentation. Lee et al. (2021) remove some unnecessary Intrusive bounding boxes, and let the Bounding Box Attribution Map use high-level information to predict segmentation, which is similar to the idea of saliency. Lin et al. (2016) proposed a weakly supervised learning method based on scribble by propagate the category information of pixels from scribbles to other unlabeled pixels. Zhang et al. (2020b) adopted a weakly-supervised salient object detection method using scribble annotations, which combing edges information and scribble information to generate rich structure results. Constrained convolutional neural network (CCNN) was proposed by Pathak et al. (2015) which uses training data with image-level labels based on weakly supervised learning to convert image tags into restrictions on the distribution of labels output by CNN. To take advantage of weakly supervised labels, Papandreou et al. (2015) used bounding box and image-level labels as the labeled training data, the Expectation Maximization Algorithm (EM) is used to estimate the category of unlabeled pixels and the parameters of the CNN. A unified framework is proposed to deal with various types of weak marks: image-level marks, bounding boxes, and partial pixel marks such as scribbles from Xu et al. (2015), it clustered all super-pixels and used the maximum interval clustering method for learning.

Weakly-supervised learning also has a wide range of applications like video segmentation. There are 2 types of video object segmentation: zero-shot segmentation and one-shot video segmentation. zero-shot is belonged to unsupervised learning, which mainly uses the natural attributes of the video (such as temporal consistency, colour, etc.) as the supervision signal, learns the visual representation, and then uses the semi-supervised video target segmentation task for evaluation. Caelles et al. (2017) proposed a typical detection method which regardless of the correlation of video timing, only do frame-by-frame detection, input a frame of video, and output the target mask. For Video Salient Object Detection, Zhao et al. (2021) provided a model that use scribbles as pseudo labels to predict salience. One-shot focuses on extracting the foreground objects that people have determined, usually assuming that the first frame is given in advance. The idea like Khoreva et al. (2016) is to rely on the temporal coherence of the video, adjust based on the result of the previous frame to get the mask off the next frame. To comprehensive 2 both detection and propagation idea,Lu et al. (2020) used frame

granularity, short-term granularity, long-range granularity and whole-video granularity these 4 areas to do the weakly supervised video object segmentation.

2.2 Camouflaged Object Detection Models

Camouflaged Object Detection is unlike general object detection and salient object detection, there are obvious differences between object and background in general object detection and salient object detection. This difference can usually be easily distinguished by the human eye. In this case, the Camouflaged Object dataset is a very challenging dataset in which the image's background and its foreground have very high similarities: objects are camouflaging, which may require a model to have better performance to detect since methods that use low-level features can not work very well. Fan et al. (2020) provided a SINet(Search and Identification Net) which contains a search Module to imitate the receptive field of the human visual system and an Identification Module for accurate detection of camouflaged targets. Though The detection of the camouflaged object is different from the general object detection, it is similar to the salient object detection. The segmentation in the camouflaged object is a foreground and Background segmentation problem and the saliency object detection can divide the input image into salient objects and background, and the camouflaged object is divided into camouflage target and background. Li et al. (2021) used the Adversarial Learning by the difference between the camouflaged objects and salient objects to find the uncertainty of the model prediction. Lv et al. (2021) introduced a triplet tasks learning model to localize, segment and rank camouflaged object ranking at the same time in order to detect different camouflaged objects.

2.3 Deep Learning Uncertainty Estimation

A trained neural network (NN) model is essentially a function with a large number of certain parameters (only addition and multiplication), no matter what input you give, it can give you an output. This makes the confidence of the model's output is very high for the clearly wrong prediction results or out of distributions inputs. Therefore, people design some approaches that model can output uncertainty to assist people who use the model to make better decisions.DeVries and Taylor (2018) proposes a new method: directly learn the confidence estimation in neural network and use this confidence estimate to determine whether the category probability is credible. At the output of the model, not only the category confidence is output, but also the confidence estimate (after sigmoid, the range is [0,1]).Kendall et al. (2016) uses dropout as an approximate inference method in BNN, so it also uses dropout as a method to obtain samples from the posterior distribution of the model. Gal and Ghahramani (2016) links this technique with the variation inference of Bernoulli distribution with weight distribution in Bayesian convolutional neural network.Compared with BNN (variational reasoning or MCMC method), Lakshminarayanan et al. (2017) simplifies the realization of the method, combine ensemble (averaging the prediction results of multiple models to obtain "model uncertainty") and adversarial training (improving local smoothness), only needs to modify the standard neural network slightly, and is suitable for distributed computing Kendall and Gal (2017) explained two kinds of uncertainty: Aleatoric uncertainty (data uncertainty) and Epistemic uncertainty (model uncertainty). Aleatoric uncertainty is actually the noise in the training data, which comes from the data collection or labeling process. These noises are random and fixed. The more noise, the greater the uncertainty of the data. It can be measured, but it cannot be reduced by increasing the data. To model and calculate Uncertainty, they introduced Monte-Carlo and Ensemble methods to model Epistemic uncertainty, and also

introduced Probabilistic Deep Learing for predicting Aleatoric Uncertainty.

Methodology

Taking advantage of our weakly labels to do the camouflaged object detection, we design a framework to successive do the training process. The first step is called Pseudo Labels Generation, the aim of this step is to generate an acceptable quality pseudo ground truth. There is 2 process in the whole generation: one is the rough generation process that aims to output roughly segmentation by using bounding box and scribble and the other is called the smooth adjustment process which can further refine the rough results by using some possible signals. To train our model, we first do the preparation process to prepare and normalize our data, then pass our input data into a network to extract features and decoders then output prediction and data uncertainty respectively. By using drop out layers, we make the model's parameters more like a distribution rather than fixed parameters. The appropriate loss function is used during the back propagation process. by handling the data uncertainty and paying more attention to the relationship between the image's pixels in our loss function, our model keeps returning more accurate results after each iteration.

3.1 Pseudo Labels Generation

To generate an acceptable pseudo label is a challenge in camouflaged objects detection since the background and foreground has a very high similarity between each other. The common traditional image segmentation methods are hard to distinguish the items from the similar colour, shape or texture. To produce an acceptable segmentation result to be our pseudo label, We divide the whole generating process into 2 steps. One step is to rely on accurate information called the rough generation process and the other is to rely on supplementary information on possibilities called the smooth adjustment process. Images are shown in Fig. 3.1

Rough generation process

Take accurate information first. To think like this: by bounding box information we can master which distinct contains our deleted objects or not. So we can easily conclude that the area inside the bounding box contains our detected objects and the outside area is always background. In this case, we apply the GrabCut Rother et al. (2004) method, which takes bounding box information and original images as input, uses GMM to get segmented masks based on graph theory. We then can get a roughly segmentation mask inside the bounding box. Scribbles are simple lines that provide some true pixel-wise information that pseudo labels can trust. Since GrabCut also supports iterative repair, which allows users to mark the foreground or background by drawing lines to get more accurate results, so the next step is to add this creditable scribbles into our mask to make our results more convincing. By passing these new pixel-wise information to our GMM model again, GMM can make the mask more accurate.

Smooth adjustment process

However, the output from the above step is still not accurate enough, we apply some supplementary information to make our mask more reliable. That is why the smooth adjustment process is produced to further refine the results. Since camouflaged object detection can be treated as a salient object detection task, introducing some possible salient signal to our mask can be helpful for GMM model to detect possible foreground. There is a salient object segmentation method called the MB+ Zhang et al. (2015), which can generate salient object segmentation using unsupervised methods. By setting a threshold, we can get some positions that may be the foreground, this information is then grabbed by the GMM model which is built from the GrabCut method. After introducing some possible salient information. The next step is to use DenseCRF Philipp and Koltun (2012) to enable the image to be segmented as far as possible at the edges.

Finally, by using rough generation process and smooth adjustment process, we apply median blur to reduce noise, then we can get our final pseudo labels. See alg. 1 for more details.

Algorithm 1 Generate Pseudo Labels

Require: original image: x, bounding box: x_{bbox}, scribble: x_{scri}
Ensure: $mask$, y
  for x in X **do**
    generate x_{mb+} by MB+ using x
    for $pixel_x$ in x **do**
      if $pixel_{scri} == 1$ **then**
        $pixel_{mask} = 1$ (1 means foreground)
      else
        $pixel_{mask} = 2$ (2 means possible background)
      end if
      if $pixel_x \in x_{bbox}$ and $pixel_{mb+} == 1$ and $pixel_{mask} == 0||2$ **then**
        $pixel_{mask} = 3$ (3 means possible foreground)
      else
        $pixel_{mask} = pixel_{mask}$ (not change)
      end if
      if $pixel_x \notin x_{bbox}$ **then**
        $pixel_{mask} = 3$ (0 means background)
      else
        $pixel_{mask} = pixel_{mask}$ (not change)
      end if
    end for
    $y = \text{grabCut}(mask)$
    $y = \text{DenseCRF}(y)$
    $y = \text{medianBlur}(y)$
  end for

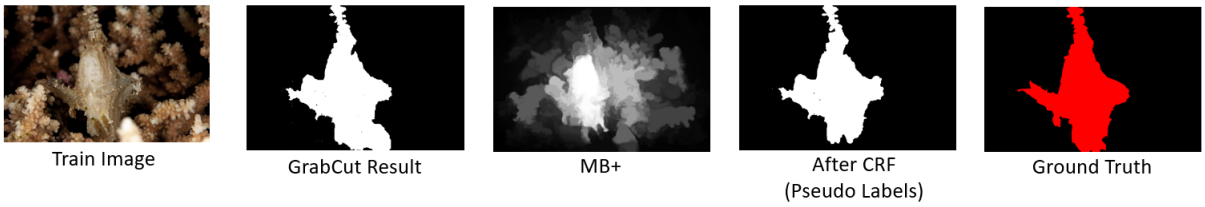

Figure 3.1: Different images related to pseudo labels, the second image is generated by GrabCut. The third image is created by MB+. The fourth image is our pseudo label which processed by CRF.

3.2 Network Structure

Training Preparation

In this paper, we used 3040 original RGB images in COD10K as our training input. By using our pseudo labels generation method, we combine corresponding bounding box information and scribbles together, generate 3040 pseudo labels as our training ground truth. To make our methods more clearly to read, we define our training dataset as:$D = \left\{x_i, y_i\right\}_{i=1}^{N}$, x_i is a input image and y_i is our pseudo labels of corresponding image, N is the total length of our training dataset. Before start training our network, we resize our input images and pseudo label to 480*480*3 and 480*480, read input images in RGB and pseudo labels in gray. Normalization and common data augmentation is also used in our network to improve the quality of our training dataset.

Features Extractor Encoder

We take ResNet50 as our backbone but make some small changes. First, we resize and normalize our images $X_{i=1}^{N}$ and then send these inputs to our network: these inputs will go through a convolutional layer, a batch normalization layer, a ReLu layer and a max pooling layer. After the pooling layer there are 4 layers connect with each other : Layer1, Layer2, Layer3 and Layer4 in the rest of our ResNet backbone which can generate different kinds of features $x = \left\{x_1, x_2, x_3, x_4\right\}$ with different size: 256x64x64, 512x32x32, 1024x16x16 and 2048x8x8. Different size of features contain different spatial information and semantic information.

Prediction Decoder and Aleatoric Decoder

It is true that no wonder how accurate the generating pseudo labels methods are, there must exist errors caused by the difference between people's pseudo labels and ground truth labels, we called this kind of error data uncertainty or aleatoric uncertainty. Aleatoric uncertainty is inevitable in weakly supervised learning and how to handle this uncertainty can be good access to improve model's performance. So, unlike other neural networks that only output predictions, our decoders can output prediction and aleatoric uncertainty at the same time. Suppose our models' parameters is θ, our model is f, and aleatoric uncertainty σ^2, then we can define our prediction as $f(x, \theta)$.

Also, to make our network have Bayesian network property, we apply Monte Carlo Drop out and set drop out layers on different kinds of features $x = \left\{x_1, x_2, x_3, x_4\right\}$ before moving forward to the next operations. For each decoder, we first set 4 different convolutional layers for different features

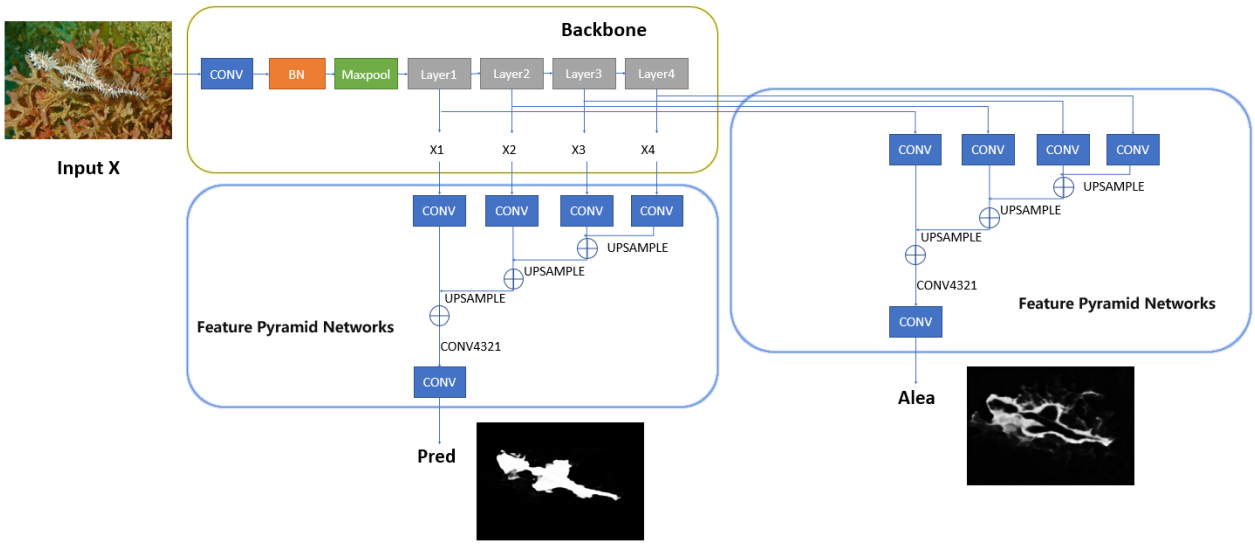

Figure 3.2: The whole structure of our network

x: conv1, conv2, conv3 and conv4 to output $x^{'} = \left\{x_1^{'}, x_2^{'}, x_3^{'}, x_4^{'}\right\}$. To the greatest extent utilize different types of features $x^{'}$, we refer to the structure of Feature Pyramid Networks, which upsample the deeper layer's features and fuse with the features from the previous layer to generate new features. That is, we can upsample $x_4^{'}$ twice and fuse it with $x_3^{'}$ to generate $x_{43}^{'}$, then upsample $x_{43}^{'}$ twice and fuse with $x_2^{'}$ to get $x_{432}^{'}$ and finally return $x_{4321}^{'}$ by fusing $x_{432}^{'}$ and $x_1^{'}$. We take $x_{4321}^{'}$ which is the Fusion of feature maps with strong low-resolution semantic information and feature maps with weak high-resolution semantic information but rich spatial information Liu et al. (2020). After that, we set a block of the fully convolutional layer that contains interpolate operation and activation function to reduce the channels of features and output our result. Since 2 decoders have the same network structure but are independent of each other, the parameters' values in the different decoders are also different. We can finally get model's prediction as $f(x, \theta)$ and aleatoric uncertainty σ^2 respectively. The whole network structure is shown in Fig 3.1. To visualize the prediction and aleatoric uncertainty, we normalize these outputs and scale them from [0,1] to [0,255]. As camouflaged object detection task only has 2 class: background and foreground, the black color means the background in prediction and the darker the color in uncertainty image, the lower the uncertainty is. The opposite reason for white color in prediction and bright color in uncertainty image.

3.3 Loss function

Fixed Aleatoric Loss

Let's begin with data loss caused by the pseudo labels first. Since the pseudo labels we provided it's not accurate enough, another extra decoder is used to output the uncertainty of our predictions. As camouflaged object detection is a classification problem,then the likelihood of prediction is $p(y|x, \theta) = sigmoid(\frac{f(x,\theta)}{exp(\sigma^2)})$. Here is the original aleatoric loss function (a L2 loss between the ground truth and the predictions and a regular term) Kendall and Gal (2017):

$$L_{alex} = \frac{1}{N}\sum_{i=1}^{N}\frac{||y_i - f(xi)||^2}{2\sigma(x_i)^2} + \frac{1}{2}log(\sigma(x_i)^2)$$

For classification problem, we don't use $||y_i - f(xi)||^2$ L2 loss but use binary cross-entropy loss L_{ce}.

To avoid the outputs of the network to be NaN by $\sigma^2 = 0$, we let the network outputs $log(\sigma^2)$ instead of σ^2. However, common binary cross-entropy loss calculates the loss of each pixel independently, ignoring the image global structure, pixel position aware loss can be a good choice to enhance structure information especially in weakly supervised detection Wei et al. (2019). By definition, L_{ppa} contains 2 parts, one is L_{wbce} and the other is L_{wiou}. In our network, L_{ppa} is

$$L_{wbce} = -\frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(1+\gamma\alpha_{ij})\sum_{l=0}^{1}(g_{ij}^{s}=l)logPr(P_{ij}^{s}=l|\Phi)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\gamma\alpha_{ij} * exp(exp(log(\sigma^2)))}$$

where

$$\alpha_{ij}^{s} = |\frac{\sum_{m,n\in A_{ij}} gt_{mn}^{s}}{\sum 1} - gt_{ij}^{s}|$$

is the weight shows how important this pixel is and γ is a hyperparameter. This weighted binary cross-entropy loss helps models be more focus on salient pixel in camouflaged object. There is no change on L_{wiou} part:

$$L_{wiou} = 1 - \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}(gt_{ij}^{s} * p_{ij}^{s}) * (1+\gamma * \alpha_{ij}^{s})}{\sum_{i=1}^{H}\sum_{j=1}^{W}(gt_{ij}^{s} + p_{ij}^{s} - gt_{ij}^{s} * p_{ij}^{s}) * (1+\gamma\alpha_{ij}^{s})}$$

The reason why we keep regular term is because if only $\sigma(x_i)^2$ is removed, the network will tend to predict all L_{ce} or L_{ppa} very large. By adding a regular term, our network learns how to output aleatoric uncertainty. Here is the final aleatoric loss:

$$L_{fixedAlex} = L_{ppa} + \frac{1}{2}log(\sigma(x_i)^2) = L_{wbce} + L_{wiou} + \frac{1}{2}log(\sigma(x_i)^2)$$

Bayesian Aleatoric Loss

In $L_{fixedAlex}$ above, $\sigma^2(x)$ is decided by some fixed model parameters. Since our network is a Bayesian network, there is another way to describe the aleatoric uncertainty by using the mean entropy:

$$U_a = E_{p(\theta|D)[H(y|x,\theta)]} = -\frac{1}{N}\sum_{n=1}^{N} H(y|x,\theta_n)$$

, where n is the sample times we made in our network by Monte Carlo Drop out. In this case, we freeze the gradient during test but let drop out layer keep working, then we can calculate the bayesian aleatoric uncertainty using entropy by using $H(y|x,\theta) = -p(y|x,\theta)log(p(y|x,\theta))$. Then, we proposal a bayesian aleatoric loss which connect the fixed aleatoric uncertainty $\sigma^2(x)$ and our baysian aleatoric uncertainty by using cross-entropy loss.

$$L_{bayAlea} = L_{bce}(\sigma^2(x), U_a)$$

By comparing these 2 different aleatoric uncertainty, we can encourage $\sigma^2(x)$ to be more likely with the sampling result so the model will be more likely to predict prediction that learn from a distribution rather than some fixed paramters during the testing process.

Smoothness Loss

Generally speaking, the optical flow of an object is relatively smooth, so the edge in the image should be basically the same as the edge in the original image. To make our prediction more smooth on

camouflaged area, we apply smooth loss to improve model's ability to find more related edges for camouflaged object Bontonou et al. (2019):

$$L_{smo} = \sum_{u,v} \sum d \in x, y\phi(|\alpha_d s_{u,v}|exp(-\alpha|\alpha_d(G * I_{u,v})|))$$

, where $|\alpha_d s_{u,v}|$ represents the smoothness of the optical flow, the smaller the value, the smoother the optical flow. Sine we only want the optical flow inside the camouflaged object to be as smooth as possible, so there is a coefficient $exp(-\alpha|\alpha_d(G * I_{u,v})|)$. At the edge of the object, the coefficient is very small, and inside the object, the coefficient is close to 1.

Total Loss

After combining these loss: Here is our final loss function:

$$L = L_{fixedAlex}(f(x|\theta), \sigma^2(x), y) + \lambda_1 L_{bayAlea}(\sigma^2(x), H(y|x)) + \lambda_2 L_{smo}(f(x|\theta), x)$$

, where x is our input image, y is our pseudo ground truth, $f(x|\theta)$ is our model's prediction, $\sigma^2(x)$ is our aleatoric uncertainty and $H(y|x)$ is the mean entropy. λ_1 and λ_2 are the hyperparamters and we set 0.4 ,0.3 to them respectively.

Results and Analysis

4.1 Setup

In our model, we used 3040 images in COD10K as our training dataset including original RGB image, objects bounding box information and scribbles. To test our dataset, we used 250 images in CAMO dataset Le et al. (2019), 75 images in CHAMELEON dataset, 2026 images in COD10K Lv et al. (2021) and 4121 images in NC4K Fan et al. (2020).

Training Details

We train our model in NVIDIA GeForce RTX 2080 GPU, set epoch amounts as 50 and batch size for each Iteration as 8. We use Adam as our optimizer ,set decay rate as 0.9. The learning rate we use in our model is $2.5e^{-4}$. It cost machine nearly 10 minutes to run a epoch and 500 minutes in total. To further improve our network, we use multi scale to scale our training images size 0.75 times, 1 time and 1.25 times for data augmentation proposal and enhance model's scale-immutability.

Evaluation Metrics

There are 4 main evaluation metrics to measure the ability of a salient object detection model nowadays: Smeasure $S_\alpha \uparrow$ Fan et al. (2017), meanFm $F_\beta^{mean} \uparrow$ Achanta et al. (2009), meanEm $E_\xi^{mean} \uparrow$ Fan et al. (2018) and MAE $M \downarrow$ Perazzi et al. (2012). Smeasure aims to study how to evaluate the foreground map. It consists of a region-oriented structural similarity measure and an object-oriented structural similarity measure. FmeasureAchanta et al. (2009), as the weighted harmonic average of Precision and Recall, has non-negative weight, which is a good substitute for P-R curve. Emeasure Fan et al. (2018)is to combine local pixels with image-level sensitization to form image-level statistics and pixel-level matching information together.MAE Perazzi et al. (2012) is to calculate the error between the continuous saliency graph and ground truth. These salient object detection metrics are widely used in camouflaged object detection.

Completed Methods

There are so many salient object detection methods but few of them can provide experiment results in camouflaged object dataset. However, there still some networks that are open-source with code and already used in camouflaged object detection task like SINet Fan et al. (2020) and RankNet Lv et al. (2021). However, network like SINet Fan et al. (2020) trained its model with 3040 images in COD10K Lv et al. (2021) and 1000 images in CAMO Le et al. (2019), that is, the total size of training dataset is 4040. To make the experiment results more comparable with different networks, we compare these completed methods with our 3040 images in COD10K Lv et al. (2021).

4.2 Performance Comparison

Compare with Baseline

Before showing the results from different completed networks, we also train 2 baseline models one is to use real ground truth to do the fully-supervised learning and the other is to use pseudo labels generated by GrabCut to do the weakly supervised learning. Here is the result Tab. 4.1.

	CAMO				CHAMELEON				COD10K				NC4K			
Method	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$
Baseline [GT]	0.696	0.560	0.732	0.107	0.737	0.625	0.782	0.089	0.794	0.713	0.876	0.042	0.810	0.746	0.884	0.046
Baseline [GrabCut]	0.660	0.536	0.741	0.122	0.675	0.564	0.759	0.110	0.703	0.607	0.804	0.070	0.729	0.656	0.828	0.076
Ours	0.673	0.550	0.754	0.120	0.688	0.578	0.775	0.107	0.712	0.626	0.814	0.068	0.739	0.675	0.837	0.073

Table 4.1: Comparing our model with 2 baseline models, one is using fully-supervised learning and the other is weakly-supervised learning

By comparing our model's result with a weakly-supervised baseline, we can see the results on all evaluation metrics are better than just using a common network and common pseudo labels to train a model. However, there is still some distance between our model with a fully-supervised baseline. By looking at the Fig. 4.1 which shows some of the results from our model, it is obvious that our model can outline the camouflaged object roughly. The aleatoric uncertainty also shows that our model is try to handle the uncertainty from the pseudo label, which also demonstrates the importance of handing the data uncertainty.

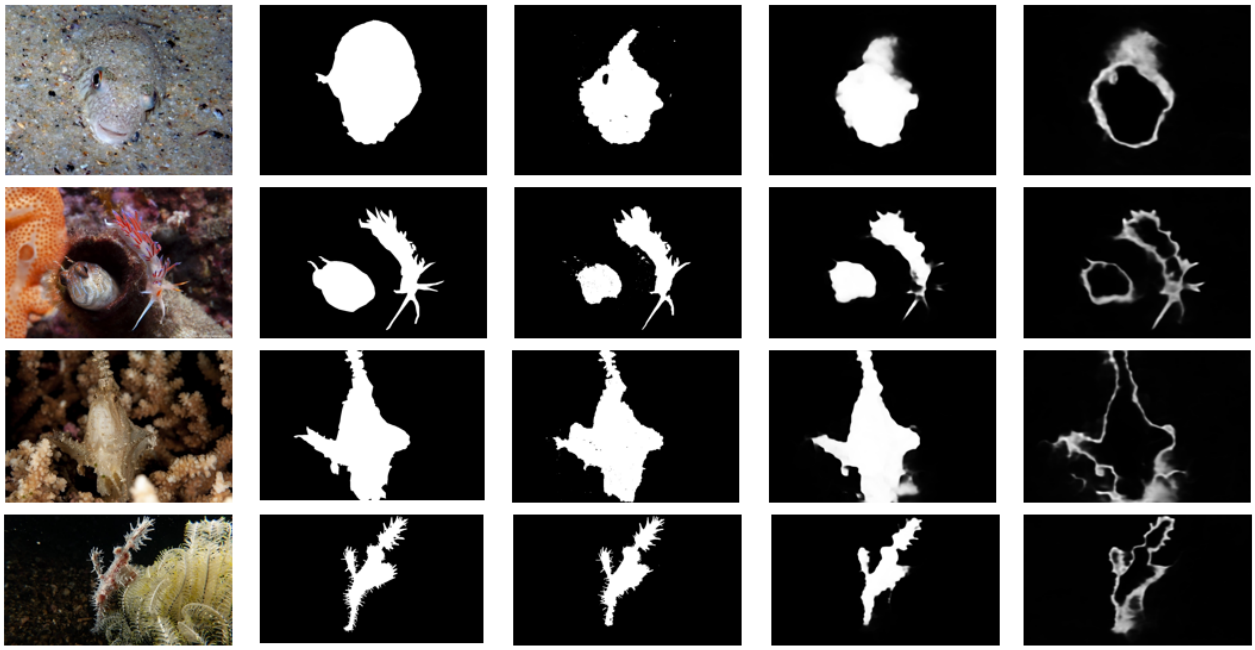

Figure 4.1: Some of the results of our model's prediction. The first column represents original input, the second column represents the ground truth, the third column represents the image's pseudo label, the fourth column represents the prediction of our output and the last column represents the uncertainty.

It can be seen that aleatoric uncertainty often occurs on the predictions outer contour, which means this kind of uncertainty can be helpful if our pseudo labels have a similar shape with ground truth. That is, the aleatoric uncertainty only can fix limited errors caused by the pseudo labels, if

there are too many differences between the pseudo labels and ground truth, the prediction will not recover to the real ground as expected.

Compare with Completed Methods

As we said in Completed Methods section, we take the experiment from Lv et al. (2021) in which these networks like SCRN Wu et al. (2019), CSNet Gao et al. (2020),UCNet Zhang et al. (2020a),BASNet Qin et al. (2019), BASNet Qin et al. (2019), SINet Fan et al. (2020) and RankedNet Lv et al. (2021) are trained by 3040 images in COD10K. See Tab. 4.2 for details.

We have to admit that these completed methods are great, even our baseline model using fully-supervised learning can not take some benefits from the comparison. Since our baseline can not compare with some of the results, that means our baseline can not really learn from the ground truth. It may be the problem of encoder, which can not provide good features for decoders, so learning how to fuse different kinds of features can also be a new idea to solve the problem.

Here are some output predictions from the test dataset Fig. 4.2. Prediction results show that our network is able to predict the rough outline but still is hard to predict edge clearly. Another main reason is that our pseudo labels generation method is not a perfect method. When the foreground and background contrast are low and confusing (like the RGB image at the second column in Fig. 4.2), or when the foreground and background have a large number of small hollow areas (like the RGB image at the fourth column in Fig. 4.2), it is easy to make mistakes.

Figure 4.2: Some of the results of our model's prediction. The first row represents original input, the second row represents the ground truth, the third row represents the prediction of our output.

	CAMO				CHAMELEON				COD10K				NC4K			
Method	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$
SCRN[Wu et al. (2019)]	0.702	0.632	0.731	0.106	0.822	0.726	0.833	0.060	0.756	0.623	0.793	0.052	0.793	0.729	0.823	0.068
CSNet[Gao et al. (2020)]	0.704	0.633	0.753	0.106	0.819	0.759	0.859	0.051	0.745	0.615	0.808	0.048	0.785	0.729	0.834	0.065
UCNet[Zhang et al. (2020a)]	0.703	0.640	0.740	0.107	0.833	0.781	0.890	0.049	0.756	0.650	0.823	0.047	0.792	0.751	0.854	0.065
BASNet[Qin et al. (2019)]	0.644	0.578	0.588	0.143	0.761	0.657	0.797	0.080	0.640	0.579	0.713	0.072	0.724	0.648	0.780	0.089
SINet[Fan et al. (2020)]	0.697	0.579	0.693	0.130	0.820	0.731	0.835	0.069	0.733	0.588	0.768	0.069	0.779	0.696	0.800	0.086
RankedNet[Lv et al. (2021)]	0.708	0.645	0.755	0.105	0.842	0.794	0.896	0.046	0.760	0.658	0.831	0.045	0.797	0.758	0.854	0.061
Ours	0.673	0.550	0.754	0.120	0.688	0.578	0.775	0.107	0.712	0.626	0.814	0.068	0.739	0.675	0.837	0.073

Table 4.2: Comparing with different network methods

	CAMO				CHAMELEON				COD10K				NC4K			
Method	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$	$S_\alpha \uparrow$	$F_\beta^{mean} \uparrow$	$E_\xi^{mean} \uparrow$	$M \downarrow$
M1	0.574	0.536	0.641	0.195	0.576	0.618	0.779	0.085	0.696	0.489	0.712	0.081	0.758	0.639	0.773	0.088
M2	0.661	0.544	0.732	0.121	0.680	0.575	0.756	0.109	0.705	0.626	0.804	0.070	0.734	0.674	0.832	0.074
M3	0.673	0.550	0.754	0.120	0.688	0.578	0.775	0.107	0.712	0.626	0.814	0.068	0.739	0.675	0.837	0.073

Table 4.3: Comparing with different pseudo labels' generation method

4.3 Ablation Study

To test the quality of our pseudo labels, we try other different generation methods and finally choose the best one by comparing. To better introduce our methods to generate pseudo labels, we name each method M1, M2 and M3 to distinguish from each other. M1 is a simple method to directly use bounding box rectangle Papandreou et al. (2015) as the pseudo label. Because M1 does not use scribble information, the result is not ideal as expected. M2 is to apply grabCut methods first, then use scribbles to further improve the segmentation results. This method contains both bounding box level labels and scribbles but is restricted by the performance of the GMM model: it can not works well on some challenging images in which foreground and background have very high similarity. The M3 method is the way we propose in this paper, both information is used in our model and we provide extra possible salient pixels then use DenseCRF to further improve. By comparing these 3 methods in Tab. 4.3

Different segmentation results are shown with Fig 4.3. By combing Tab. 4.3 and Fig 4.3, it is obvious that the method we proposed in our paper achieved a better performance.

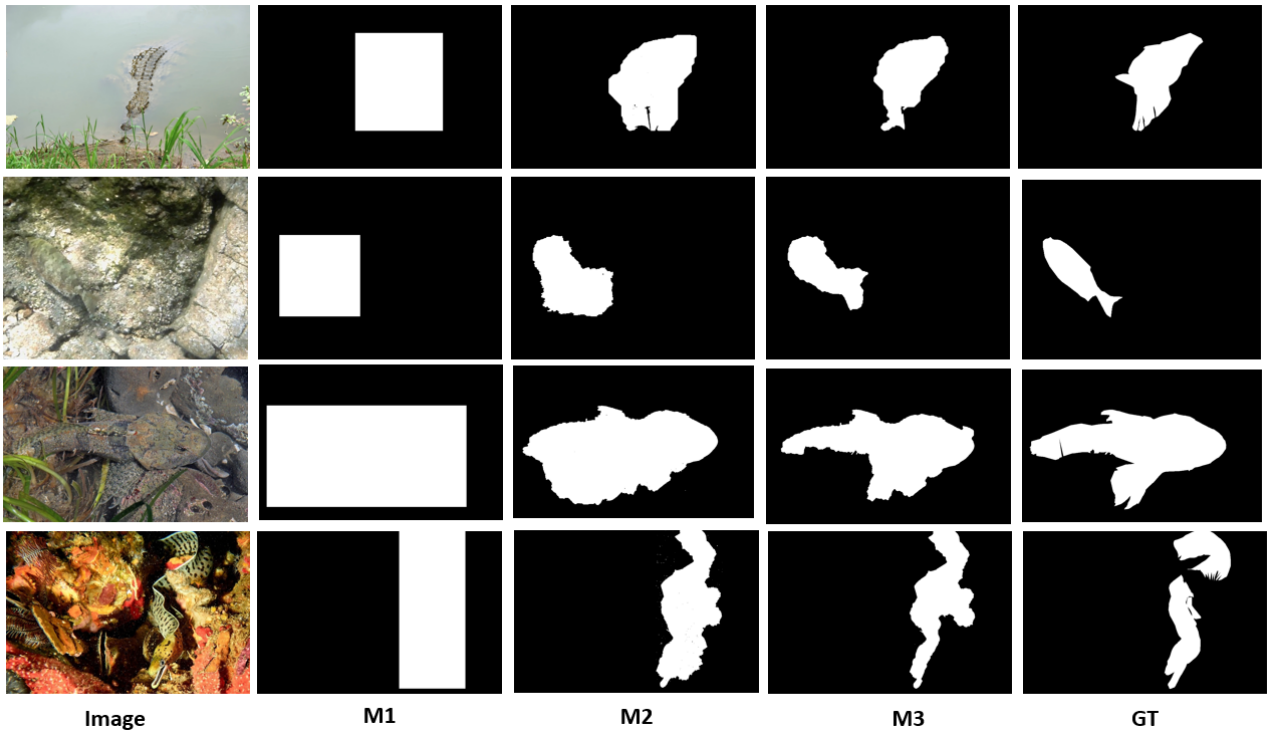

Figure 4.3: Comparisons of different generated pseudo methods.M1,M2,M3 referred to bounding box rectangle, GrabCut and our pseudo label generation method

Conclusions and Future Development

In this paper, we present a new end-to-end weakly supervised camouflaged object detection framework based on Bayesian theory. By reasonable using 2 weakly annotation labels, we propose a new method to generate pseudo labels. Though our pseudo labels are not accurate enough, we design a network that can simultaneously output prediction and aleatoric uncertainty. The experiment result proves the effectiveness of loss function and enhances our model's performance in a way. It is a pity that the final result of our model can not compare with the state of art. In future work, we should design another powerful feature extractor to further improve the ability of the encoder. What is more, generating pseudo labels process costs a lot of time, finding an efficient method to simplify the current method can improve applicability. Finally, changing the loss function to encourage model to boldly predict more uncertainty outside the object rather than just focus on the area inside the object can be a vital point. Looking forward to seeing an improved model in the future :)

Bibliography

Achanta, R.; Hemami, S.; Estrada, F.; and Süsstrunk, S., 2009. Frequency-tuned salient region detection. In *CVPR*, CONF, 1597–1604.

Bontonou, M.; Lassance, C.; Hacene, G. B.; Gripon, V.; Tang, J.; and Ortega, A., 2019. Introducing graph smoothness loss for training deep learning architectures.

Caelles, S.; Maninis, K.-K.; Pont-Tuset, J.; Leal, L.; Cremers, D.; and Gool, L. V., 2017. One-shot video object segmentation.

Chum, L.; Subramanian, A.; Balasubramanian, V. N.; and Jawahar, C. V., 2019. Beyond supervised learning: A computer vision perspective. *Journal of the Indian Institute of Science*, 99 (2019), 177–199.

DeVries, T. and Taylor, G. W., 2018. Learning confidence for out-of-distribution detection in neural networks.

Fan, D.-P.; Cheng, M.-M.; Liu, Y.; Li, T.; and Borji, A., 2017. Structure-measure: A new way to evaluate foreground maps. 4548–4557.

Fan, D.-P.; Gong, C.; Cao, Y.; Ren, B.; Cheng, M.-M.; and Borji, A., 2018. Enhanced-alignment measure for binary foreground map evaluation. 698–704.

Fan, D.-P.; Ji, G.-P.; Sun, G.; Cheng, M.-M.; Shen, J.; and Shao, L., 2020. Camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gal, Y. and Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning.

Gao, S.-H.; Tan, Y.-Q.; Cheng, M.-M.; Lu, C.; Chen, Y.; and Yan, S., 2020. Highly efficient salient object detection with 100k parameters.

Hsu, C.-C.; Hsu, K.-J.; Tsai, C.-C.; Lin, Y.-Y.; and Chuang, Y.-Y., 2019. Weakly supervised instance segmentation using the bounding box tightness prior. In *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc. `https://proceedings.neurips.cc/paper/2019/file/e6e713296627dff6475085cc6a224464-Paper.pdf`.

Kendall, A.; Badrinarayanan, V.; and Cipolla, R., 2016. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding.

Kendall, A. and Gal, Y., 2017. What uncertainties do we need in bayesian deep learning for computer vision?

Khoreva, A.; Perazzi, F.; Benenson, R.; Schiele, B.; and Sorkine-Hornung, A., 2016. Learning video object segmentation from static images.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles.

Le, T.-N.; Nguyen, T. V.; Nie, Z.; Tran, M.-T.; and Sugimoto, A., 2019. Anabranch network for camouflaged object segmentation. *Computer Vision and Image Understanding*, 184 (Jul 2019), 45–56. doi:10.1016/j.cviu.2019.04.006. http://dx.doi.org/10.1016/j.cviu.2019.04.006.

Lee, J.; Yi, J.; Shin, C.; and Yoon, S., 2021. Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation.

Li, A.; Zhang, J.; Lv, Y.; Liu, B.; Zhang, T.; and Dai, Y., 2021. Uncertainty-aware joint salient object and camouflaged object detection.

Lin, D.; Dai, J.; Jia, J.; He, K.; and Sun, J., 2016. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation.

Liu, Y.; Zhang, X.; Zhang, B.; and Chen, Z., 2020. Deep network for road damage detection. In *2020 IEEE International Conference on Big Data (Big Data)*, 5572–5576. doi:10.1109/BigData50022.2020.9377991.

Lu, X.; Wang, W.; Shen, J.; Tai, Y.-W.; Crandall, D.; and Hoi, S. C. H., 2020. Learning video object segmentation from unlabeled videos.

Lv, Y.; Zhang, J.; Dai, Y.; Li, A.; Liu, B.; Barnes, N.; and Fan, D.-P., 2021. Simultaneously localize, segment and rank the camouflaged objects.

Papandreou, G.; Chen, L.-C.; Murphy, K.; and Yuille, A. L., 2015. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation.

Pathak, D.; Philipp; and Darrell, T., 2015. Constrained convolutional neural networks for weakly supervised segmentation.

Perazzi, F.; Krähenbühl, P.; Pritch, Y.; and Hornung, A., 2012. Saliency filters: Contrast based filtering for salient region detection. 733–740.

Philipp and Koltun, V., 2012. Efficient inference in fully connected crfs with gaussian edge potentials.

Qin, X.; Zhang, Z.; Huang, C.; Gao, C.; Dehghan, M.; and Jagersand, M., 2019. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Rother, C.; Kolmogorov, V.; and Blake, A., 2004. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23, 3 (Aug. 2004), 309–314. doi:10.1145/1015706.1015720. https://doi.org/10.1145/1015706.1015720.

Wei, J.; Wang, S.; and Huang, Q., 2019. F3net: Fusion, feedback and focus for salient object detection.

Wu, Z.; Su, L.; and Huang, Q., 2019. Stacked cross refinement network for edge-aware salient object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Xu, J.; Schwing, A. G.; and Urtasun, R., 2015. Learning to segment under various forms of weak supervision. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3781–3790. doi:10.1109/CVPR.2015.7299002.

Zhang, J.; Fan, D.-P.; Dai, Y.; Anwar, S.; Saleh, F. S.; Zhang, T.; and Barnes, N., 2020a. Uc-net: Uncertainty inspired rgb-d saliency detection via conditional variational autoencoders.

Zhang, J.; Sclaroff, S.; Lin, Z.; Shen, X.; Price, B.; and Měch, R., 2015. Minimum barrier salient object detection at 80 fps. In *IEEE International Conference on Computer Vision(ICCV)*.

Zhang, J.; Yu, X.; Li, A.; Song, P.; Liu, B.; and Dai, Y., 2020b. Weakly-supervised salient object detection via scribble annotations.

Zhao, W.; Zhang, J.; Li, L.; Barnes, N.; Liu, N.; and Han, J., 2021. Weakly supervised video salient object detection.