Contour Knowledge Transfer for Salient Object Detection

Xin Li[1*], Fan Yang[1*], Hong Cheng[1], Wei Liu[1], Dinggang Shen[2]

[1] University of Electronic Science and Technology of China, Chengdu 611731, China
[2] Department of Radiology and BRIC, University of North Carolina at Chapel Hill, North Carolina 27599, USA
{xinli_uestc,fanyang_uestc}@hotmail.com, hcheng@uestc.edu.cn, dgshen@med.unc.edu

Abstract. In recent years, deep Convolutional Neural Networks (CNNs) have broken all records in salient object detection. However, training such a deep model requires a large amount of manual annotations. Our goal is to overcome this limitation by automatically converting an existing deep contour detection model into a salient object detection model without using any manual salient object masks. For this purpose, we have created a deep network architecture, namely Contour-to-Saliency Network (C2S-Net), by grafting a new branch onto a well-trained contour detection network. Therefore, our C2S-Net has two branches for performing two different tasks: 1) predicting contours with the original contour branch, and 2) estimating per-pixel saliency score of each image with the newly-added saliency branch. To bridge the gap between these two tasks, we further propose a contour-to-saliency transferring method to automatically generate salient object masks which can be used to train the saliency branch from outputs of the contour branch. Finally, we introduce a novel alternating training pipeline to gradually update the network parameters. In this scheme, the contour branch generates saliency masks for training the saliency branch, while the saliency branch, in turn, feeds back saliency knowledge in the form of saliency-aware contour labels, for fine-tuning the contour branch. The proposed method achieves state-of-the-art performance on five well-known benchmarks, outperforming existing fully supervised methods while also maintaining high efficiency.

Keywords: Saliency detection · deep learning · transfer learning

1 Introduction

Salient object detection, which aims at locating the most visually conspicuous object(s) in natural images, is critically important to computer vision. It can be used in a variety of tasks such as human pose estimation [5], semantic segmentation [11], image/video captioning [25], and dense semantic correspondences [34].

* Both authors contribute equally to this work.
Code and pre-trained models are available at https://github.com/lixin666/C2SNet.

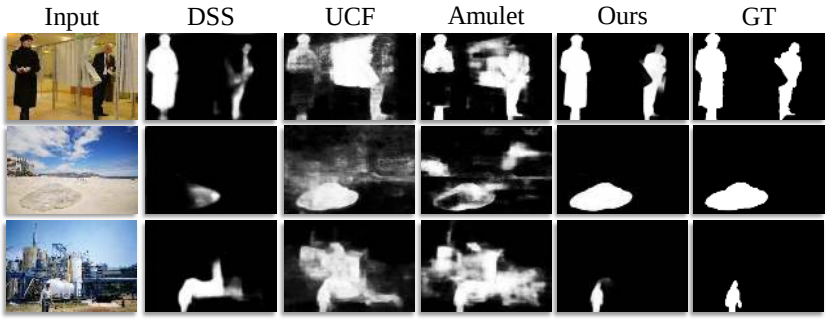


Fig. 1. Saliency maps produced by currently best deep saliency models (DSS [8], UCF [38], and Amulet [37]) and ours. Different from these fully supervised methods, our method requires *no groundtruth salient object mask* for training deep CNNs.

Over the past decades, the techniques of salient object detection have evolved dramatically. Traditional methods [3, 4, 20] only use low-level features and cues for identifying salient regions in an image, leading to their inability to summarize high-level semantic knowledge. Therefore, these methods are unsuitable for handling images with complex scenes. Recently, fully-supervised approaches [8, 9, 21, 24] based on deep Convolutional Neural Networks (CNNs) have greatly improved the performance of salient object detection. The success of these methods depends mostly on a huge number of training images containing manually annotated salient objects. Unfortunately, in salient object detection, annotations are provided in the form of pixel-wise masks. Annotating a large-scale training dataset requires tremendous cost and effort.

To eliminate the need for time-consuming image annotation, we propose to facilitate feature learning in salient object detection by borrowing knowledge from an existing contour detection model. Although salient object detection and contour extraction seem inherently different, they are actually related to each other. On one hand, contours provide useful priors or cues for identifying salient regions in an image. For example, salient regions are often surrounded by contours. On the other hand, saliency knowledge helps remove background clutter, and thus improves contour detection results. Therefore, it is reasonable to transfer knowledge between these two involved domains [16–18].

Our goal is to convert a trained contour detection model (CEDN) [35] into a saliency detection model *without* using any manually labeled salient object masks. With this goal, we first graft a new branch onto the existing CEDN to form a multi-task network architecture, i.e., Contour-to-Saliency Network (C2S-Net). Then, we employ the well-trained contour branch to generate contour maps for all images and use a novel contour-to-saliency transferring method to produce the corresponding saliency masks. The newly-added branch is trained under the strong supervision of these automatically generated saliency masks. After that, the trained branch in turn transfers the learned saliency knowledge,

in the form of saliency-aware contour labels, to the contour branch. In this way, the original contour branch learns to detect the contours of only the most attention-grabbing object(s). The interaction between the original branch and newly-added branch is iterated in order to increase accuracy. Although the generated salient object masks and saliency-aware contour labels may contain errors in the beginning, they gradually become more reliable after several iterations. More importantly, the well-trained CEDN undergoes essential changes through the alternating training procedure between the two branches (i.e., Contour-to-Saliency procedure and Saliency-to-Contour procedure), becoming a powerful saliency detection model, where one branch focuses on salient object contour detection and the other branch predicts saliency score of each pixel.

Despite not using manually annotated salient object labels for training, our proposed method is capable of generating a reliable saliency map for each input (See Fig. 1). The experiments show that our proposed method yields higher accuracy than the existing fully-supervised deep models. Furthermore, it takes only 0.03 second to perform each image, which is much faster than most existing methods.

In summary, this paper makes the following three major contributions:

- We present a new idea and solution for salient object detection by automatically converting a well-trained contour detection model into a saliency detection model, *without* requiring any groudtruth salient object labels.
- We propose a novel Contour-to-Saliency Network (C2S-Net) based on the well-trained contour detection network. In this architecture, the same feature encoder is used by both the original contour branch and the newly-added saliency branch. We also introduce cross-domain connections to enable the saliency branch to fully encode contour knowledge during the learning process.
- We introduce a simple yet effective contour-to-saliency transferring method to bridge the gap between contours and salient object regions. Therefore, the results generated by the well-trained contour branch can be used to generate reliable saliency masks for training the saliency branch. In addition, we propose a novel alternating training pipeline to update the network parameters of our C2S-Net.

2 Related Work

Salient object detection has evolved quickly over the past two decades. Earlier methods [3, 4, 20] rely on low-level features and cues such as intensity, color, and texture. Although these methods can produce accurate saliency maps in most simple cases, they are unable to deal with complex images due to the lack of semantic knowledge.

In recent years, fully-supervised CNNs have demonstrated highly accurate performance in salient object detection tasks. These methods can be categorized into two groups: region-based methods and pixel-wise saliency prediction methods. Region-based methods predict saliency score in a region-wise manner.

Zhao *et al.* [39] integrate both global and local context into a multi-context CNN framework for saliency detection. In [13], a multi-layer fully connected network is proposed for estimating the saliency score of each super pixel. Wang *et al.* [28] proposed the integration of both local estimation and global search for patch-wise saliency score estimation. All these methods treat image patches as independent units, and thus they may result in spatial information loss and redundant computations. To overcome these drawbacks, pixel-wise saliency prediction methods directly map an input image to the corresponding saliency map by using a trained deep Fully Convolutional Network (FCN). Li *et al.* [19] proposed the use of a multi-task fully-convolutional neural network for salient object detection. Wang *et al.* [30] proposed a recurrent FCN to encode saliency prior knowledge for salient object detection. In [8], Hou *et al.* introduce short connections into the Holistically-nested Edge Detector (HED) network architecture [31] so as to solve scale-space problems in salient object detection. Li *et al.* [21] developed a multi-scale cascade network, which can encode multi-scale context information and thus produce a better result. In general, these fully-supervised CNN-based methods can achieve good performance even when handling complex scenes. However, training deep CNN models requires a large amount of pixel-level annotations, which have to be created manually in a time-consuming and expensive way.

Notable previous attempts at detecting salient object(s), while using no saliency mask for training, are Weakly Supervised Saliency (WSS) [29] and Supervision by Fusion (SBF) [37] methods. WSS takes advantage of image-level tags to generate pixel-wise annotations for training a deep saliency model. SBF trains the desirable deep saliency model by automatically generating reliable supervisory signals from the fusion process of weak saliency models. However, due to the lack of detailed object shape information, these methods perform far worse in challenging cases compared to fully-supervised methods. Compared with the methods proposed in [29, 37], our method can achieve much higher accuracy. This is because our solution obviates the need for image-level tags in training, and thus the accuracy can be increased by using a much larger number of training images from any class (not limited to predefined categories). Furthermore, the contour knowledge is successfully transferred for salient region detection. This enables the deep CNN network to learn detailed object shape information and improve the overall performance. To the best of our knowledge, the idea of transferring contour knowledge for salient object detection has not been investigated before.

3 Approach

3.1 Overview

This paper tackles the problem of borrowing contour knowledge for salient object detection without the need of labeled data. Given an existing contour detection network (CEDN) [35], our objective is to convert this already well-trained model

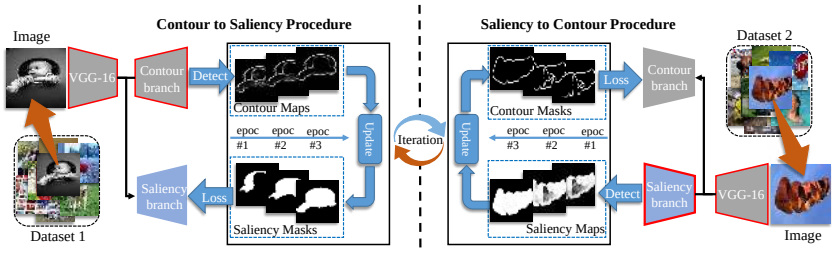


Fig. 2. The proposed alternating training pipeline. Our training algorithm is composed of two procedures: (a) contour-to-saliency procedure and (b) saliency-to-contour procedure. In the contour-to-saliency procedure, we use the generated saliency masks to train the newly-added saliency branch. In the saliency-to-contour procedure, the generated saliency-aware contours are used to fine-tune the original contour branch.

into an accurate deep saliency detection model without using any manually labeled saliency mask.

First, we propose a novel Contour-to-Saliency Network by grafting a new branch onto the existing CEDN. In this architecture, the original contour branch and the newly-added saliency branch share the same feature extractor (or encoder). The feature extractor and contour branch are initialized using CEDN, and the saliency branch is randomly initialized. Therefore, our C2S-Net has the ability to naturally detect contours of the input image after parameter initialization.

Then, we train the saliency branch and update the parameters of the contour branch on two different unlabeled image sets through a novel alternating training pipeline. The training algorithm is composed of two procedures: 1) contour-to-saliency procedure and 2) saliency-to-contour procedure. In the contour-to-saliency procedure, the contour branch is first used to detect contours in each image. Next, a novel contour-to-saliency transfer method is utilized to generate salient object masks based on the detected contours. These generated masks are used to simulate strong supervision over the saliency branch. In the saliency-to-contour procedure, we employ the opposite process to update the parameters of the contour branch. Alternating the two procedures above enables the saliency branch to progressively derive semantically strong features for salient object detection, and the contour branch learns to identify only the contours of salient regions. Fig. 2 illustrates the main steps of the alternating training pipeline. In the following sections, we will give a detailed description of C2S-Net, contour-to-saliency transfer method, and our alternating training pipeline.

3.2 Contour-to-Saliency Network

Architecture. Fig. 3 illustrates the detailed configuration of our Contour-to-Saliency Network (C2S-Net). Our C2S-Net is rooted in a fully Convolutional

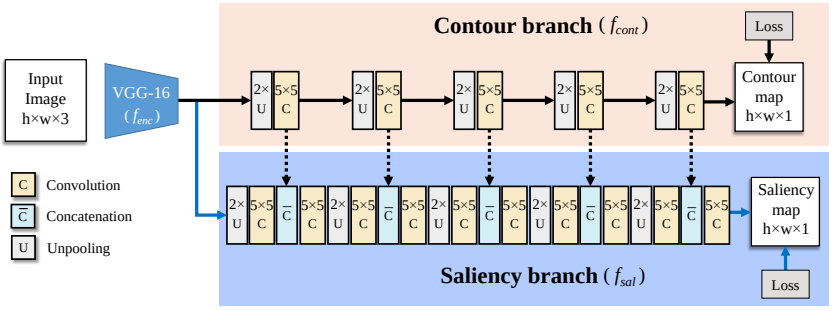


Fig. 3. The two-branch C2S-Net roots in the CEDN [35] for salient object detection. With cross-domain connections (the dashed line), the saliency branch is naturally capable of consolidating both saliency and contour knowledge.

Encoder-Decoder Network (CEDN) [35] originally designed for contour detection. We update the network by grafting a new decoder for saliency detection onto the original encoder. By doing this, our C2S-Net is made of three major components: encoder (f_{enc}), contour decoder (f_{cont}) and saliency decoder (f_{sal}). In our network, the encoder extracts high-level feature representations from an input image, the contour decoder identifies contours of the salient region, and the saliency decoder estimates the saliency score of each pixel.

Encoder. The encoder takes an image $\mathcal{I}_i$ as its input, and outputs a feature map $\mathcal{F}_i$. Following CEDN, we employ VGG-16 [27] for feature extractor part (encoder f_{enc}) with the last two layers removed.

Contour Decoder. The contour decoder is built upon the feature extractor, and it takes a feature map $\mathcal{F}_i$, and produces a saliency-aware contour map $C(\mathcal{F}_i, \theta_c)$ where θ_c denotes the model parameter of contour branch. The training of contour decoder can be treated as a per-pixel regression problem to the ground-truth contour labels, by minimizing the following objective function:

$$\min_{\theta_c} \sum_i e_{cont}(\mathcal{L}_{cont}(\mathcal{I}_i), C(\mathcal{F}_i; \theta_c)), \tag{1}$$

where $\mathcal{L}_{cont}(\mathcal{I}_i)$ denotes the ground-truth contour labels of the i-th example, and $e_{cont}(\mathcal{L}_{cont}(\mathcal{I}_i), C(\mathcal{F}_i; \theta_c))$ is the per-pixel loss function.

Saliency Decoder. The saliency decoder f_{sal} share the same encoder f_{enc} with the contour decoder f_{enc}. Similarly, it takes the feature map $\mathcal{F}_i$ as input and produces a single-channel saliency map $S(\mathcal{F}_i, \theta_s)$, where θ_s is the model parameter of saliency decoder. Because salient object detection is a more difficult task than contour detection, we add another convolutional layer in each saliency decoder group. The objective of the saliency branch is to minimize the per-pixel error between the ground-truths and estimated saliency maps. Formally, the objective function can be written as:

$$\min_{\theta_s} \sum_i e_{sal}(\mathcal{L}_{sal}(\mathcal{I}_i), S(\mathcal{F}_i; \theta_s)), \tag{2}$$

where $\mathcal{L}_{sal}(\mathcal{I}_i)$ is the ground-truth salient object mask of the *i-th* image, and $e_{sal}(\mathcal{L}_{sal}(\mathcal{I}_i), C(\mathcal{F}_i;\theta_s))$ is the per-pixel loss of $S(\mathcal{F}_i;\theta_s))$ with respect to $\mathcal{L}_{sal}(\mathcal{I}_i)$.

Cross-Domain Connections. In order to make full use of contour information, we introduce cross-domain connections into our C2S-Net to enable the saliency branch to encode contour knowledge as well.

Specifically, in the saliency decoder stage, the feature learning of the second convolutional layer encodes both the learned features $f_{s_i}^{cont}$ from contour branch and the convolutional features $f_{s_i}^{sal}$ of its previous layer. Therefore, the second convolutional feature map $\tilde{f}_{s_i}^{sal}$ on the *i-th* level in the saliency branch is formally written as:

$$\tilde{f}_{s_i}^{sal} = \sigma(cat(f_{s_i}^{cont}, f_{s_i}^{sal}) \otimes w_{s_i}^{sal} + b_{s_i}^{sal}), \tag{3}$$

where $w_{s_i}^{sal}$ and $b_{s_i}^{sal}$ are convolutional filters and biases for the *i-th* decoder stage in the saliency branch, respectively. $\otimes$ represents convolution operation, and $cat(\cdot)$ is used to concatenate the two learned feature maps of different tasks. RELU serves as the non-linear function $\sigma(\cdot)$.

Our C2S-Net use pixel-level saliency-aware contour labels $\mathcal{L}_{cont}$ and saliency masks $\mathcal{L}_{sal}$ as supervision. Unlike the fully supervised methods, in this paper, these labels are automatically generated, rather than manually annotated. This is achieved by a novel transferring method, which will be introduced in the following section.

3.3 Contour-to-Saliency Transfer

Since our C2S-Net is rooted in a well-trained contour detection network [35], its contour branch is able to identity contours after parameter initialization. The detected contours provide important cues for salient object detection. As observed by many previous works [6, 7], salient objects are usually well-surrounded by contours or edges. Therefore, we can leverage this important cue to bridge the gap between object contours and salient object regions.

With detected contour maps in a large collection of unlabeled images, our goal is to utilize them to generate corresponding salient object masks, so as to simulate strong human supervision over saliency branch training. First, we adopt Multiscale Combinatorial Grouping (MCG) [1] to generate some proposal candidate masks $\mathcal{C}$ *from our detected contours* in each image. Then, different from [2], we design an objective function to pick out only a very few masks $\mathcal{B}$ from $\mathcal{C}$ that are most likely to cover the entire salient regions to form the salient object mask $\mathcal{L}_{sal}$ for each image. Formally, our objective function is defined as:

$$\begin{aligned} &\max_{\mathcal{B}} \{S(\mathcal{B}) - \alpha \cdot O(\mathcal{B}) - \kappa \cdot N(\mathcal{B})\} \\ &\quad s.t. \quad \mathcal{B} \subseteq \mathcal{C} \end{aligned} \tag{4}$$

where $S(\cdot)$ is the data term that encourages the selection of region proposals with a higher saliency score. $O(\cdot)$ denotes the overlap term which penalizes intersection between selected region proposals. $N(\cdot)$ is number term which penalizes the number of selected region masks. α and κ are the weights of overlap term and

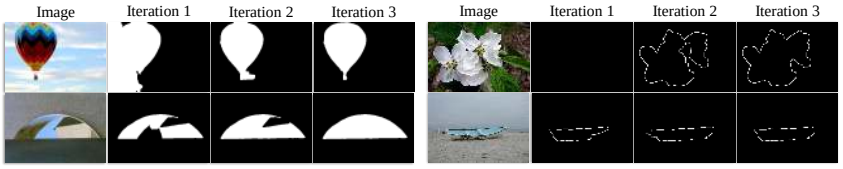


Fig. 4. Update of contour labels and saliency masks. Here we show the generated (a) saliency masks and (b) contour labels in Iter #1, Iter #2 and Iter #3. These updated labels and masks will be used as the supervision for the next iteration.

number term, respectively. By maximizing the objective function above, we can determine a small number of region proposals whose union serves as the salient object mask $\mathcal{L}_{sal}$ used for training.

To be more specific, a binary variable c_i is used to indicate the selection of proposal b_i from all candidate masks $\mathcal{C}$. If b_i is selected, we set $c_i = 1$ otherwise $c_i = 0$. Therefore, we rewrite Eq.4 as follows:

$$\begin{aligned} &\max\{\sum_{b_i \subseteq C} S_i c_i - \alpha \cdot \sum_{\substack{b_i, b_j \in C \\ i \neq j}} K(b_i, b_j) c_i c_j - \kappa \cdot \sum_{b_i \subseteq C} c_i\} \\ &s.t. \quad c_i, c_j = 0 \quad or \quad 1 \end{aligned} \tag{5}$$

Here, $K(b_i, b_j)$ is the Intersection-over-Union (IoU) score between two different region masks b_i and b_j. S_i denotes the score reflecting the likelihood of region mask b_i to be a salient region mask. According to [6, 7], a region that is better surrounded by contours is more likely to be a salient region. In addition, the saliency map obtained in the previous stage provides useful prior knowledge. Therefore, we also use it to estimate the saliency score of a given region mask. Formally, the saliency score of each region proposal can be formally written as:

$$S_i = K(cnt(b_i), C^{e_r}) + \gamma \cdot K(b_i, S^{e_r}) \tag{6}$$

where $cnt(b_i)$ denotes a function that extracts contour map from a given region mask b_i. This is simply achieved through computing the gradient on the binary region mask b_i. C^{e_r} and S^{e_r} denote the detected contour and saliency map after the *r-th* training epoch, respectively. As the parameters of saliency branch are randomly initialized and our network cannot generate saliency maps at the very beginning, we set the combination weight $\gamma = 0$ in the first epoch, and $\gamma = 1$ in the following epoches.

Optimization. Seeking the solution to Eq. 5 is a NP-hard problem. Here, we adopt a greedy algorithm described in [36] to address this problem efficiently.

3.4 Alternating Training

Our C2S-Net has three important components: encoder f_{enc}, contour decoder f_{cont} and newly-added saliency decoder f_{sal}. We initialize parameters of both

f_{enc} and f_{cont} by parameter values of the existing well-trained contour detection model (CEDN) [35], and initialize parameters of f_{sal} randomly from the normal distribution. To avoid the poor local optimum problem, we use two different sets of unlabeled images ($\mathcal{M}$ and $\mathcal{N}$) to interactively train the saliency branch and contour branch. During the training time, the network parameters are optimized by back-propagation and stochastic gradient descent (SGD).

We iteratively perform contour-to-saliency procedure and saliency-to-contour procedure, fixing one set of network parameters while solving for the other set. Specifically, in the contour-to-saliency procedure, by fixing the encoder parameters θ_e and the contour decoder parameters θ_c, we generate contour map of each image on the unlabeled set $\mathcal{M}$ by using the initialized C2S-Net in the first time-step (and the updated C2S-Net in each following time-step). After that, we use the proposed contour-to-saliency transfer method to produce salient object masks $\mathcal{L}_{sal}$ as training samples for updating the saliency decoder parameters θ_s. In this procedure, we also measure confidence score of every generated contour map by $\frac{C(\mathcal{F}_i,\theta_c)\geq 0.9}{C(\mathcal{F}_i,\theta_c)\geq 0.1}$, and choose contour maps whose scores are larger than a predefined threshold ($\vartheta = 0.15$) so as to filter out unreliable contour maps. In the saliency-to-contour procedure, we fix the network parameters θ_e and θ_s , and use the learned C2S-Net to generate both contour maps and saliency maps. These generated results are then utilized to produce salient object masks on unlabeled set $\mathcal{N}$ using Eq. 5. We adopt $cnt(\cdot)$ in Eq. 6 to generate saliency-aware contour labels $\mathcal{L}_{cont}$, and use these generated labels to update the contour decoder parameters θ_c. For each round of iteration, we update the network parameters to improve the quality of estimated labels for the next round.

Our alternating training pipeline successfully takes advantage of the complementary benefits of two related domains. On one hand, the contour branch is able to learn saliency knowledge, and thus it can focus more on the contours of those attention-grabbing objects. More importantly, the training samples generated by saliency branch are not limited to a small number of predefined categories. Therefore, the contour branch can learn saliency properties from a large set of images to detect contours of "unseen" objects. On the other hand, the saliency branch learns detailed object shape information so that it can produce saliency maps with clear boundaries. As shown in Fig. 4, the estimated salient object masks and contour maps become more and more reliable, and then provide useful information for network training.

4 Experiments

4.1 Experimental Setup

Dataset. The training set contains $10K$ images from MSRA10K (ignoring labels), and another $20K$ unlabelled images collected from the Web as additional training data. These images contain one or multiple object(s) and cluttered backgrounds, and are not overlapped with any test image. We randomly divide the training set into two subsets, $\mathcal{M}$ and $\mathcal{N}$, to train contour branch and

saliency branch of our C2S-Net, respectively. In addition, we augment each subset through horizontal flipping.

For the performance evaluation, we utilize five most challenging benchmarks including ECSSD [32], PASCAL-S [22], DUT-OMRON [33], HKU-IS [14] and DUTS-TE [29].

Implementation. Our C2S-Net is implemented based on the public code of CEDN [35], which was based on Caffe toolbox [10]. The network parameters of encoder and contour decoder are initialized by the CEDN model. The parameters of saliency decoder are initialized randomly. We set $\alpha = 0.5$ and $\kappa = 0.25$ in Eq. 4.

During training, we adopt the "poly" learning rate policy, where the learning rate is automatically scaled by $(1 - \frac{iter}{max_{iter}})^p$. We set the initial learning rate to 10^{-6}, and p to 0.9. The maximum number of iterations is set based on the number of training data ($max_{iter} = N \times 3$, where N denotes the number of training data). The mini-batch size is set to 5. At each training round, we update network parameters by fine-tuning the model trained from the previous round. In addition, as discussed in Sec. 3.4, at each training round, we first solve for parameters of one branch while fixing the parameters of the other, and then perform the opposite procedure.

During testing, the input RGB image is forwarded through our C2S-Net to generate a saliency map with the same size as the output. Unlike other methods, we *do not* need to adopt any pre-processing or post-processing steps, e.g., DenseCRF, for further improving the detected results.

Evaluation Metrics. We use four evaluation metrics to evaluate the performance of our method: Precision-Recall curves (PR), F-measure (F_β), weighted F-measure (F_β^w), and Mean Absolute Error (MAE). The F-measure is computed by $F_\beta = (1 + \beta^2)\frac{Precision \times Recall}{\beta^2 Precision + Recall}$, where β^2 is set to 0.3 to emphasize precision. We also adopt the weighted F-measure [26] to assess the performance of our method, which is defined as $F_\beta^w = (1 + \beta^2)\frac{Precision^w \times Recall^w}{\beta^2 Precision^w + Recall^w}$. MAE is defined as the average pixel-wise absolute difference between the ground-truth mask and estimated saliency map. All these universally-agreed evaluation metrics have been widely adopted by previous works.

4.2 Ablation Analysis

In this section, we conduct ablation studies on ECSSD dataset by comparing the weighted F-measure (F_β^w) and MAE to verify impact of each component in the framework. Details of the results are summarized in Tab.1.

Impact of Cross-Domain Connections. We evaluate the performance differences of the proposed C2S-Net with and without cross-domain connections (CDC). For a fair comparison, we train both models using the same training images (i.e., $5K$ images randomly selected from MSRA10K with pixel-wise ground truths), and the same training parameters which are described in Sec. 4.1. The experiments show that our C2S-Net with CDC can improve the F_β^w by 2.4%, and significantly lower the MAE score by 21.3%. Compared with only sharing the

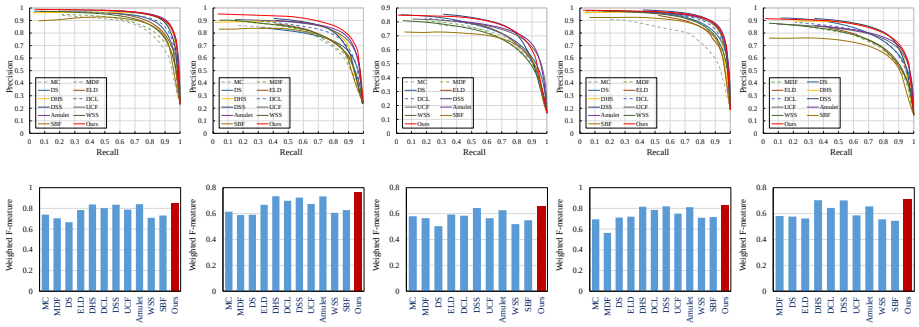
Table 1. Analysis of the proposed method. Our results are obtained on ECSSD. "CDC" denotes the cross domain connections that used in our C2S-Net. "AVG-P" means the two-stage strategy, "WTA" denotes the "winner-take-all" strategy, and "CTS" refers to the contour-to-saliency transferring method used in this paper. "SCJ" denotes that we optimize the parameters of two branches jointly, and "$AT_{(i)}$" means that *i-th* alternating training iterations are used to update network parameters. "† " denotes the model used in this paper for comparing with fully supervised models. Weighted F-measure (F_β^w): the higher the better; MAE: the lower the better.

Method	data/annotations	F_β^w	MAE
C2S-Net	5K w/ masks	0.793	0.103
C2S-Net + CDC	5K w/ masks	0.812	0.081
C2S-Net + CDC + AVG-P	5K w/o masks	0.665	0.121
C2S-Net + CDC + WTA	5K w/o masks	0.732	0.112
C2S-Net + CDC + CTS	5K w/o masks	0.743	0.093
C2S-Net + CDC + CTS + SCJ	10K w/o masks	0.759	0.088
C2S-Net + CDC + CTS + $AT_{(1)}$	10K w/o masks	0.778	0.080
C2S-Net + CDC + CTS + $AT_{(3)}$	10K w/o masks	0.837	0.059
C2S-Net + CDC + CTS + $AT_{(5)}$	10K w/o masks	0.838	0.059
C2S-Net + CDC + CTS + $AT_{(3)}$	20K w/o masks	0.849	0.056
† C2S-Net + CDC + CTS + $AT_{(3)}$	30K w/o masks	0.852	0.054

same encoder, our CDC enables the proposed model to better explore the intrinsic correlations between saliency detection and contour detection, and results in a better performance.

Effectiveness of Contour-to-Saliency Transferring. Automatically generating a reliable salient object mask for each image, based on generated proposal candidate masks $\mathcal{C}$ (about 500 proposals), is a challenging task. Here, we take three different approaches to generate saliency masks for training our model. One approach is the two-stage strategy, the second is the "winner-take-all" strategy, and the third is our contour-to-saliency transferring strategy. These approaches are respectively referred to as AVG-P, WTA, and CTS. Specifically, for AVG-P, we first simply take an average of all proposals (generated from detected contours) to form a saliency map for each image, and then use SalCut [3] to produce its salient object mask. As for WTA, all generated proposals are re-scored according to Eq. 6 and only the proposal with the highest score is picked out to serve as salient object mask for each image. As for our CTS, we use the method described in Sec. 3.3 to produce salient object masks for all images. We also use the same $5K$ images from MSRA10K as the training set, but we ignore all of the manual masks. The third, fourth and fifth lines of Tab.1 show the corresponding results of using AVG-P, WTA and CTS to generate saliency masks for training our C2S-Net, respectively. Clearly, the proposed CTS enables our C2S-Net to achieve much better performance than other strategies.

Impact of Alternating Training. To verify the effectiveness of our alternating training (AT) approach, we use another $5K$ unlabeled images, the remaining images of MSRA10K, to serve as the training set of contour branch. The experiments show that our alternating training approach (AT) can largely boost the performance of our C2S-Net. After the first iteration, our model achieves com-



(a) ECSSD (b) PASCAL-S (c) DUT (d) HKU-IS (e) DUTS-TE

Fig. 5. From top to bottom, Precision-recall (PR) curves and *weighted* F-measure of our method and other state-of-the-art approaches are shown, respectively.

petitive performance as fully-supervised approaches ($F_\beta^w = 0.778$, and $MAE = 0.080$). Our C2S-Net with three AT iterations achieves much better performance according to F_β^w (0.837) and MAE score (0.059). We observe that the performance of our model with five AT iterations ($F_\beta^w = 0.838$, and $MAE = 0.059$) is just slightly better than that of model with three AT iterations. This is because the estimated saliency masks and contour maps have already become reliable enough after three AT iterations. Considering the training time and model's performance, we believe that three AT iterations should be a good choice.

In addition, to show the superiority of our alternating training scheme, we use the same $10K$ images with estimated labels (including both saliency and contour labels) to train our C2S-Net. One loss is for contour branch and another loss is for saliency branch. We optimize the parameters of two branches jointly, and denote this training strategy as SCJ in Tab. 1. According to our experiments, when given the same amount of training data, our alternating training strategy can achieve much better performance.

Impact of Data Size. According to our reported results (Tab.1), the models performance on ECSSD improves as the training data expands. This indicates that data size is a big influencing factor for saliency model's performance. Feeding more training samples to the deep CNN models can lead to better performance.

4.3 Comparison to Other Methods

We compare the proposed method with nine top-ranked fully deep supervised saliency detection models including MC [39], MDF [13], DS [19], ELD [12], DHS [23], DCL [15], DSS [8], UCF [38], and Amulet [37], one weakly supervised deep saliency model WSS [29], and one unsupervised deep saliency model SBF [37]. In all experiments, we use the models provided by original authors.

Quantitative Comparison. In order to obtain a fair comparison with existing weakly supervised and unsupervised deep models, we first use the same training

Table 2. Quantitative comparisons with 10 leading CNN-Based methods on five widely-used benchmarks. The top three results are shown in Red, Blue, and Green, respectively. F_β: the higher the better; MAE: the lower the better.

Methods	ECSSD		PASCAL-S		DUT		HKU-IS		DUTS-TE	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
SBF [37]	0.852	0.880	0.765	0.130	0.685	0.108	0.842	0.075	0.698	0.107
WSS [29]	0.856	0.103	0.770	0.139	0.689	0.110	0.860	0.079	0.737	0.100
Ours(10K)	0.896	0.059	0.835	0.086	0.733	0.079	0.883	0.051	0.790	0.066
MC [39]	0.822	0.107	0.721	0.147	0.703	0.088	0.781	0.098	-	-
MDF [14]	0.832	0.105	0.759	0.142	0.694	0.092	0.860	0.129	0.768	0.099
DS [19]	0.882	0.122	0.757	0.172	0.716	0.120	0.866	0.079	0.776	0.090
ELD [12]	0.869	0.098	0.777	0.121	0.720	0.091	0.767	0.071	0.758	0.097
DHS [23]	0.902	0.061	0.820	0.092	-	-	0.892	0.052	0.812	0.065
DCL [15]	0.887	0.072	0.798	0.109	0.718	0.094	0.879	0.059	0.771	0.079
DSS [8]	0.903	0.062	0.821	0.101	0.761	0.074	0.899	0.051	0.813	0.064
UCF [38]	0.910	0.078	0.819	0.127	0.735	0.132	0.885	0.074	0.771	0.117
Amulet [37]	0.915	0.059	0.828	0.100	0.743	0.098	0.895	0.052	0.778	0.085
Ours(30K)	0.910	0.054	0.846	0.081	0.757	0.071	0.896	0.048	0.807	0.062

set as in SBF [37] (MSRA10K without masks), and test over all of the evaluation datasets using the same model. As shown in Tab. 2, our model (with $10K$ training images) consistently outperforms the exsiting weakly supervised and unsupervised deep saliency models with a large margin, and compares favorably with the top-ranked fully supervised deep models.

One of the advantages of our method is that it can use a large amount of unlabeled data for training, while the existing fully supervised methods are constrained by the amount of labeled data. Here, we use additional 20K unlabelled images collected from the Web (30K in total) to train our model, and compare it with all top-ranked fully deep supervised models. As shown in Tab. 2 and Fig. 5, our method can largely outperform other leading methods in nearly all evaluation metrics across all the datasets. Specifically, on ECSSD, PASCAL-S, DUT-OMRON, HKU-IS, and DUTS-TE, our method decreases the lowest MAE score by 8.5%, 11.9%, 4.1%, 5.9% and 3.1%, respectively. This indicates that our method can produce more confident results and generate more reliable saliency maps that are close to the groud truth. In terms of F-measure and PR curves, our method consistently ranks among the top three on all datasets (see Tab. 2 and Fig. 5). In addition, as shown in Fig. 5, we improve the current best weighted F-measure (F_β^w) by 1.2%, 4.4%, 2.7%, 0.1% and 0.2% on ECSSD, PASCAL-S, DUT-OMRON, HKU-IS, and DUTS-TE, respectively. In general, the experimental results convincingly demonstrate the effectiveness of our method. It also should be noted that our method requires *no manual salient object label* for training the network while other top-ranked deep models are trained with pixel-wise annotations. As our method can benefit from unlimited number of unlabeled images, it has full potential for further performance improvement.

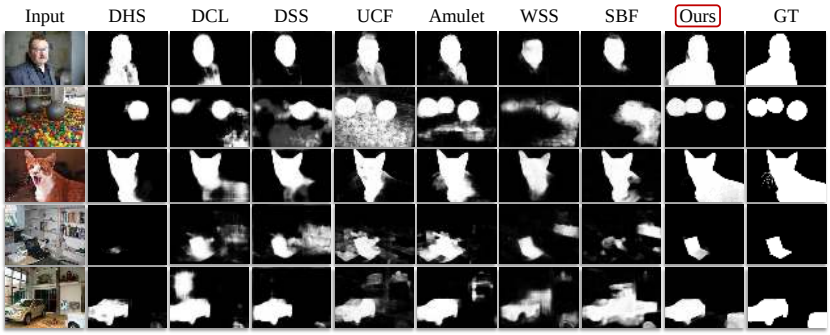


Fig. 6. Qualitative comparisons of our method and the state-of-the-art approaches. The ground truth (GT) is shown in the last column.

Table 3. Comparison of running times.

Method	MC	MDF	DS	ELD	DHS	DCL	DSS	UCF	Amulet	WSS	SBF	**Ours**
Times(s)	2.38	8.04	0.73	0.59	0.06	1.17	0.05	0.11	0.06	0.02	0.03	0.03

Qualitative Comparison. Fig. 6 provides a qualitative comparison between our method and other approaches. It can be seen that our method can consistently and accurately highlight the salient objects in different challenging cases. Because the contour knowledge has been encoded by our C2S-Net, our model can always better preserve object contours than other comparison methods.

Speed Performance. Lastly, we show the speed performance of our method and other approaches in Tab. 3. The evaluation is conducted with an NVIDIA GTX 1080ti GPU with 11G RAM. Our method takes only 0.03 second to produce a saliency map for a 400×300 input image.

5 Conclusions

In this paper, we propose a novel method to borrow contour knowledge for salient object detection. We first build a C2S-Net by grafting a new branch onto a well-trained object contour detection network. To bridge the gap between contours and salient object regions, we propose a novel transferring method that can automatically generate a saliency mask for each image from its contour map. These generated masks are then used to train the saliency branch of C2S-Net. Finally, we use a novel alternating training pipeline to further improve the performance of our C2S-Net. Extensive experiments on five datasets show that our method surpasses the current top saliency detection approaches.

Acknowledgments. This research was funded in part by the National Key R&D Program of China (2017YFB1302300), the National Nature Science Foundation of China (U1613223), and the Open Research Subject of Comprehensive Health Management Center of Xihua University (JKGL2018-029).

References

1. Arbelez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. pp. 328–335 (2014)
2. Bertasius, G., Shi, J., Torresani, L.: Semantic segmentation with boundary neural fields. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
3. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.M.: Global contrast based salient region detection. TPAMI **37**(3), 569–582 (2015)
4. Cheng, M.M., Warrell, J., Lin, W.Y., Zheng, S., Vineet, V., Crook, N.: Efficient salient region detection with soft image abstraction. In: ICCV. pp. 1529–1536 (2013)
5. Chu, X., Yang, W., Ouyang, W., Ma, C., Yuille, A.L., Wang, X.: Multi-context attention for human pose estimation. In: CVPR (July 2017)
6. Deng, Q., Luo, Y.: Edge-based method for detecting salient objects. Optical Engineering **50**(5), 301–301 (2011)
7. Du, S., Chen, S.: Salient object detection via random forest. IEEE SPL **21**(1), 51–54 (2013)
8. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.H.S.: Deeply supervised salient object detection with short connections. In: CVPR (July 2017)
9. Hu, P., Shuai, B., Liu, J., Wang, G.: Deep level sets for salient object detection. In: CVPR (July 2017)
10. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM MM. pp. 675–678 (2014)
11. Jin, B., Ortiz Segovia, M.V., Susstrunk, S.: Webly supervised semantic segmentation. In: CVPR (July 2017)
12. Lee, G., Tai, Y.W., Kim, J.: Deep saliency with encoded low level distance map and high level features. In: CVPR (2016)
13. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: CVPR. pp. 5455–5463 (2015)
14. Li, G., Yu, Y.: Visual saliency based on multiscale deep features. In: CVPR. pp. 5455–5463 (2015)
15. Li, G., Yu, Y.: Deep contrast learning for salient object detection. In: CVPR. pp. 478–487 (2016)
16. Li, J., Lu, K., Huang, Z., Zhu, L., Shen, H.T.: Transfer independently together: A generalized framework for domain adaptation. IEEE Transactions on Cybernetics (2018). https://doi.org/10.1109/TCYB.2018.2820174
17. Li, J., Wu, Y., Zhao, J., Lu, K.: Low-rank discriminant embedding for multiview learning. IEEE transactions on cybernetics **47**(11), 3516–3529 (2017)
18. Li, J., Zhao, J., Lu, K.: Joint feature selection and structure preservation for domain adaptation. In: IJCAI. pp. 1697–1703 (2016)
19. Li, X., Zhao, L., Wei, L., Yang, M.H., Wu, F., Zhuang, Y., Ling, H., Wang, J.: Deepsaliency: Multi-task deep neural network model for salient object detection. TIP **25**(8), 3919–3930 (2016)
20. Li, X., Yang, F., Chen, L., Cai, H.: Saliency transfer: An example-based method for salient object detection. In: IJCAI. pp. 3411–3417 (2016)
21. Li, X., Yang, F., Cheng, H., Chen, J., Guo, Y., Chen, L.: Multi-scale cascade network for salient object detection. In: ACM MM (October 2017)

22. Li, Y., Hou, X., Koch, C., Rehg, J.M., Yuille, A.L.: The secrets of salient object segmentation. In: CVPR. pp. 280–287 (2014)
23. Liu, N., Han, J.: Dhsnet: Deep hierarchical saliency network for salient object detection. In: CVPR. pp. 678–686 (2016)
24. Luo, Z., Mishra, A., Achkar, A., Eichel, J., Li, S., Jodoin, P.M.: Non-local deep features for salient object detection. In: CVPR (July 2017)
25. Ramanishka, V., Das, A., Zhang, J., Saenko, K.: Top-down visual saliency guided by captions. In: CVPR (July 2017)
26. Ran, M., Zelnikmanor, L., Tal, A.: How to evaluate foreground maps. In: Computer Vision and Pattern Recognition. pp. 248–255 (2014)
27. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
28. Wang, L., Lu, H., Ruan, X., Yang, M.H.: Deep networks for saliency detection via local estimation and global search. In: CVPR. pp. 3183–3192 (2015)
29. Wang, L., Lu, H., Wang, Y., Feng, M., Wang, D., Yin, B., Ruan, X.: Learning to detect salient objects with image-level supervision. In: CVPR (July 2017)
30. Wang, L., Wang, L., Lu, H., Zhang, P., Ruan, X.: Saliency detection with recurrent fully convolutional networks. In: ECCV. pp. 825–841. Springer (2016)
31. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV. pp. 1395–1403 (2015)
32. Xie, Y., Lu, H., Yang, M.H.: Bayesian saliency via low and mid level cues. TIP **22**(5), 1689–1698 (2013)
33. Yang, C., Zhang, L., Lu, H., Xiang, R., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: CVPR. pp. 3166–3173 (2013)
34. Yang, F., Li, X., Cheng, H., Li, J., Chen, L.: Object-aware dense semantic correspondence. In: CVPR (July 2017)
35. Yang, J., Price, B., Cohen, S., Lee, H., Yang, M.H.: Object contour detection with a fully convolutional encoder-decoder network. In: CVPR (June 2016)
36. Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., Mech, R.: Unconstrained salient object detection via proposal subset optimization. In: CVPR. pp. 5733–5742 (2016)
37. Zhang, P., Wang, D., Lu, H., Wang, H., Ruan, X.: Amulet: Aggregating multi-level convolutional features for salient object detection. In: ICCV (2017)
38. Zhang, P., Wang, D., Lu, H., Wang, H., Yin, B.: Learning uncertain convolutional features for accurate saliency detection. In: ICCV (Oct 2017)
39. Zhao, R., Ouyang, W., Li, H., Wang, X.: Saliency detection by multi-context deep learning. In: CVPR. pp. 1265–1274 (2015)