In this paper, we propose the first framework (UC-Net) to employ uncertainty for RGB-D saliency detection by learning from the data labeling process. Existing RGB-D saliency detection methods treat the saliency detection task as a point estimation problem, and produce a single saliency map following a deterministic learning pipeline. Inspired by the saliency data labeling process, we propose probabilistic RGB-D saliency detection network via conditional variational autoencoders to model human annotation uncertainty and generate multiple saliency maps for each input image by sampling in the latent space. With the proposed saliency consensus process, we are able to generate an accurate saliency map based on these multiple predictions. Quantitative and qualitative evaluations on six challenging benchmark datasets against 18 competing algorithms demonstrate the effectiveness of our approach in learning the distribution of saliency maps, leading to a new state-of-the-art in RGB-D saliency detection.

1. Introduction

Object-level visual saliency detection involves separating the most conspicuous objects that attract humans from the background [27, 2, 55, 63, 38, 29, 62]. Recently, visual saliency detection from RGB-D images have attracted lots of interest due to the importance of depth information in human vision system and the popularity of depth sensing technologies [61, 64]. Given a pair of RGB-D images, the task of RGB-D saliency detection aims to predict a saliency map by exploring the complementary information between color image and depth data.

The de-facto standard for RGB-D saliency detection is to train a deep neural network using ground truth (GT) saliency maps provided by the corresponding benchmark datasets, where the GT saliency maps are obtained through human consensus or by the dataset creators [18]. Building upon large scale RGB-D datasets, deep convolutional neural network based models [21, 61, 6, 24] have made profound progress in learning the mapping from an RGB-D image pair to the corresponding GT saliency map. Considering the progress for RGB-D saliency detection under this pipeline, in this paper, we would like to argue that this pipeline fails to capture the uncertainty in labeling the GT saliency maps.

According to research in human visual perception [33], visual saliency detection is subjective to some extent. Each person could have specific preferences in labeling the saliency map (which has been previous discussed in user-specific saliency detection [26]). Existing approaches to RGB-D saliency detection treat saliency detection as a point estimation problem, and produce a single saliency map for each input image pair following a deterministic learning pipeline, which fails to capture the stochastic characteristic of saliency, and may lead to a partisan saliency model as shown in second row of Fig. 1. Instead of obtaining only a single saliency prediction (point estimation), we are interested in how the network produces multiple predictions (distribution estimation), which are then processed further to generate a single prediction in a similar way to how the GT saliency maps are created.

In this paper, inspired by human perceptual uncertainty,
we propose a conditional variational autoencoders [50] (CVAE) based RGB-D saliency detection model UC-Net to produce multiple saliency predictions by modeling the distribution of output space as a generative model conditioned on the input RGB-D images to account for the human uncertainty in annotation.

However, there still exists one obstacle before we could apply the probabilistic framework, that is existing RGB-D benchmark datasets generally only provide a single GT saliency map for each RGB-D image pair. To produce diverse and accurate predictions⁵, we resort to the “hide and seek” [49] principle following the orientation shifting theory [26] by iteratively hiding the salient foreground from the RGB image for testing, which forces the deep network to learn the saliency map with diversity. Through this iterative hiding strategy, we obtain multiple saliency maps for each input RGB-D image pair, which reflects the diversity/uncertainty from human labeling.

Moreover, depth data in the RGB-D saliency dataset can be noisy, and a direct fusion of RGB and depth information may overwhelm the network to fit noise. To deal with the noisy depth problem, a depth correction network is proposed as an auxiliary component to produce depth images with rich semantic and geometric information. We also introduce a saliency consensus module to mimic the majority voting mechanism for saliency GT generation.

Our main contributions are summarized as: 1) We propose a conditional probabilistic RGB-D saliency prediction model that can produce diverse saliency predictions instead of a single saliency map; 2) We provide a mechanism via saliency consensus to better model how saliency detection works; 3) We present a depth correction network to decrease noise that is inherent in depth data; 4) Extensive experimental results on six RGB-D saliency detection benchmark datasets demonstrate the effectiveness of our UC-Net.

2. Related Work

2.1. RGB-D Saliency Detection

Depend on how the complementary information between RGB images and depth images is fused, existing RGB-D saliency detection models can be roughly classified into three categories: early-fusion models [43], late-fusion models [54, 24] and cross-level fusion models [61, 5, 7, 6, 64]. Qu et al. [43] proposed an early-fusion model to generate feature for each superpixel of the RGB-D pair, which was then fed to a CNN to produce saliency of each superpixel. Recently, Wang et al. [54] introduced a late-fusion network (i.e. AFNet) to fuse predictions from the RGB and depth branch adaptively. In a similar pipeline, Han et al. [24] fused the RGB and depth information through fully connected layers. Chen et al. [7] used a multi-scale multi-path network for different modality information fusion. Chen et al. [5] proposed a complementary-aware RGB-D saliency detection model by fusing features from the same stage of each modality with a complementary-aware fusion block. Chen et al. [6] presented attention-aware cross-level combination blocks for multi-modality fusion. Zhao et al. [64] integrated a contrast prior to enhance depth cues, and employed a fluid pyramid integration framework to achieve multi-scale cross-modal feature fusion. To effectively incorporate geometric and semantic information within a recurrent learning framework, Li et al. [61] introduced a depth-induced multi-scale RGB-D saliency detection network.

2.2. VAE or CVAE based Deep Probabilistic Models

Ever since the seminal work by Kingma et al. [31] and Rezende et al. [45], variational autoencoder (VAE) and its conditional counterpart CVAE [50] have been widely applied in various computer vision problems. To train a VAE, a reconstruction loss and a regularizer are needed to penalize the disagreement of the prior and posterior distribution of the latent representation. Instead of defining the prior distribution as a standard Gaussian distribution, CVAE utilizes the input observation to modulate the prior on Gaussian latent variables to generate the output. In low-level vision, VAE and CVAE have been applied to the tasks such as image background modeling [34], latent representations with sharp samples [25], difference of motion modes [57], medical image segmentation model [3], and modeling inherent ambiguities of an image [32]. Meanwhile, VAE and CVAE have been explored in more complex vision tasks such as uncertain future forecast [1, 53], human motion prediction [47], and shape-guided image generation [12]. Recently, VAE algorithms have been extende to 3D domain targeting applications such as 3D meshes deformation [52], and point cloud instance segmentation [59].

To the best of our knowledge, CVAE has not been exploited in saliency detection. Although Li et al. [34] adopted VAE in their saliency prediction framework, they used VAE to model the image background, and separated salient objects from the background through the reconstruction residuals. In contrast, we use CVAE to model labeling variants, indicating human uncertainty of labeling. We are the first to employ CVAE in saliency prediction network by considering the human uncertainty in annotation.

3. Our Model

In this section, we present our probabilistic RGB-D saliency detection model based on a conditional variational autoencoder, which learns the distribution of saliency maps rather than a single prediction. Let ξ = {Xi, Yi}i=1N be the training dataset, where Xi = {Ii, Di} denotes the RGB-D

⁵Diversity of prediction is related to the content of image. Image with clear content may lead to consistent prediction (1st row in Fig. 1), while complex image may produce diverse predictions (2nd row of Fig. 1).
PredictionNet takes both stochastic features and deterministic features to produce a final saliency map. We perform saliency consensus in Y with GT obtain the majority voting of the final predictions.

Figure 3. Overview of the proposed framework during testing. We sample the PriorNet multiple times to generate diverse and accurate predictions. The saliency consensus module is then used to obtain the majority voting of the final predictions.

Input (consisting of the RGB image Iᵢ and the depth image Dᵢ), Yᵢ denotes the ground truth saliency map. The whole pipeline of our model during training and testing are illustrated in Fig. 2 and Fig. 3, respectively.

Our network is composed of five main modules: 1) LatentNet (PriorNet and PosteriorNet) that maps the RGB-D input Xᵢ (for PriorNet) or Xᵢ and Yᵢ (for PosteriorNet) to the low dimensional latent variables zᵢ ∈ R^K (K is dimension of the latent space); 2) DepthCorrectionNet that takes Iᵢ and Dᵢ as input to generate a refined depth image Dᵢ′; 3) SaliencyNet that maps the RGB image Iᵢ and the refined depth image Dᵢ′ to saliency feature maps Sᵢ′; 4) PredictionNet that employs stochastic features Sᵢ+ from LatentNet and deterministic features Sᵢd from SaliencyNet to produce our saliency map prediction Pᵢ; 5) A saliency consensus module in the testing stage that mimics the mechanism of saliency GT generation to evaluate the performance with the provided single GT saliency map Yᵢ. We will introduce each module as follows.

3.1. Probabilistic RGB-D Saliency Model via CVAE

The Conditional Variational Autoencoder (CVAE) modulates the prior as a Gaussian distribution with parameters conditioned on the input data X. There are three types of variables in the conditional generative model: conditioning variable X (RGB-D image pair in our setting), latent variable z, and output variable Y. For the latent variable z drawn from the Gaussian distribution P(z|X), the output variable Y is generated from P(Y|X,z), then the posterior of z is formulated as Q(z|X,Y). The loss of CVAE is defined as:

$$L_{CVAE} = E_{q(z|x,y)}[- \log p_w(Y|X,z)] + D_{KL}(Q(z|x,y)||P(z|x))$$ (1)

where P_w(Y|X,z) is the likelihood of P(Y) given latent variable z and conditioning variable X, the KL-Leibler Divergence D_{KL}(Q(z|x,y)||P(z|x)) works as a regularization loss to reduce the gap between the prior P(z|x) and the auxiliary posterior Q(z|x,y). In this way, CVAE aims to model the log likelihood P(Y) under encoding error D_{KL}(Q(z|x,y)||P(z|x)). Following the standard practice in conventional CVAE [50], we design a CVAE-based RGB-D saliency detection network, and describe each component of our model in the following.

**LatentNet:** We define P_th(z|X) as PriorNet that maps the input RGB-D image pair X to a low-dimensional latent feature space, where \( \theta \) is the parameter set of PriorNet. With the same network structure and provided GT saliency map Y, we define Q_φ(z|X,Y) as PosteriorNet, with \( \phi \) being the posterior net parameter set. In the LatentNet (PriorNet and PosteriorNet), we use five convolutional layers to map the input RGB-D image X (or concatenation of X and Y for the PosteriorNet) to the latent Gaussian variable \( z \sim N(\mu, \text{diag}(\sigma^2)) \), where \( \mu, \sigma \in \mathbb{R}^K \), representing the
mean and standard deviation of the latent Gaussian variable, as shown in Fig. 4.

Let us define parameter set of PriorNet and PosteriorNet as \((\mu_{\text{prior}}, \sigma_{\text{prior}})\) and \((\mu_{\text{post}}, \sigma_{\text{post}})\) respectively. The KL-Divergence in Eq. (1) is used to measure the distribution mismatch between the prior net \(P_{\theta}(z|X)\) and posterior net \(Q_{\phi}(z|X, Y)\), or how much information is lost when using \(Q_{\phi}(z|X, Y)\) to represent \(P_{\theta}(z|X)\). Typical using of CVAE involves multiple versions of ground truth \(Y\) [32] to produce informative \(z \in \mathbb{R}^{K}\), with each position in \(z\) represents possible labeling variants or factors that may cause diverse saliency annotations. As we have only one version of GT, directly training with the provided single GT may fail to produce diverse predictions as the network will simply fit the provided annotation \(Y\).

**Generate Multiple Predictions:** To produce diverse and accurate predictions, we propose an iterative hiding technique inspired by [49] following the orientation shifting theory [26] to generate more annotations as shown in Fig. 5. We iteratively hide the salient region in the RGB image with mean of the training dataset. The RGB image and its corresponding GT are set as the starting point of the “new label generation” technique. We first hide the ground truth salient object in the RGB image, and feed the modified image to an existing RGB saliency detection model [42] to produce a saliency map and treat it as one candidate annotation. We repeat salient object hiding technique three times for each training image to obtain four different sets of annotations in total (including the provided GT), and we term this dataset as “AugedGT”, which is our training dataset.

During training, different annotations (as shown in Fig. 5) in \(Q_{\phi}(z|X, Y)\) can force the PriorNet \(P_{\theta}(z|X)\) to encode labeling variants of a given input \(X\). As we have already obtained diverse annotations with the proposed hiding technique, we are expecting the network to produce diverse predictions for images with complicated context. During testing, we can obtain one stochastic feature \(S^s\) (input of the “PredictionNet”) of channel size \(K\) each time we sample as shown in Fig. 3.

**SaliencyNet:** We design SaliencyNet to produce a deterministic saliency feature map \(S^d\) from the input RGB-D data, where the refined depth data comes from the Depth-CorrectionNet. We use VGG16 [48] as our encoder, and remove layers after the fifth pooling layer. To enlarge the receptive field, we follow DenseASPP [58] to obtain feature map with the receptive field of the whole image on each stage of the VGG16 network. ”daspp” is the DenseASPP module [58].
the PredictionNet to map $S^{msd}$ to a single channel saliency map $P$. During testing, with multiple stochastic features $S^s$, we can obtain multiple predictions by sampling $S^s$ from the LatentNet $N(\mu_{\text{prior}}, \text{diag}(\sigma^2_{\text{prior}}))$ multiple times.

### 3.2. DepthCorrectionNet

Two main approaches are employed to acquire depth data for RGB-D saliency detection: through depth sensors such as Microsoft Kinect, e.g., DES [8], and NLPRI [41] datasets; or computing depth from stereo cameras, examples of such datasets are SSB [40] and NU2K [28]. Regardless of the capturing technique, noise is inherent in the depth data. We propose a semantic guided depth correction network to produce refined depth information as shown in Fig. 2, termed as “DepthCorrectionNet”. The encoder part of the DepthCorrectionNet is the same as the “SaliencyNet”, while the decoder part is composed of four sequential convolutional layers and bilinear upsampling operation.

We assume that edges of the depth map should be aligned with edges of the RGB image. We adopt the boundary IOU loss [39] as a regularizer for DepthCorrectionNet to achieve a refined depth, which is guided by intensity of the RGB image. The full loss for DepthCorrectionNet is defined as:

$$L_{\text{Depth}} = L_{\text{sl}} + L_{\text{loub}},$$

where $L_{\text{sl}}$ is the smooth $\ell_1$ loss between the refined depth $D'$ and the raw depth $D$, $L_{\text{loub}}$ is the boundary IOU loss between the refined depth $D'$ and intensity $I_g$ of the RGB image $I$. Given the predicted depth map $D'$ and intensity of RGB image $I_g$, we follow [39] to compute the first-order derivatives of $D'$ and $I_g$. Subsequently, we calculate the magnitude $|gD'|$ and $gI$ of the gradients of $D'$ and $I_g$, and define the boundary IOU loss as:

$$L_{\text{loub}} = 1 - \frac{1}{2} \frac{|gD'| \cap gI|}{|gD'| + gI|}. \quad (3)$$

### 3.3. Saliency Consensus Module

Saliency detection is subjective to some extent, and it is common to have multiple annotators to label one image, and the final ground truth saliency map is obtained through majority voting strategy [18]. Although it is well known in the saliency detection community about how the ground truth is acquired; yet, there exists no research on embedding this mechanism into deep saliency frameworks. Current models define saliency detection as a point estimation problem instead of a distribution estimation problem. We, instead, use CVAE to obtain the saliency distribution. Next, we embed saliency consensus into our probabilistic framework to compute the majority voting of different predictions in the testing stage as shown in Fig. 3.

During testing, we sample PriorNet with fixed $\mu_{\text{prior}}$ and $\sigma_{\text{prior}}$ to obtain a stochastic feature $S^s$. With each $S^s$ and deterministic feature $S^d$ from SaliencyNet, we obtain one version of saliency prediction $P$. To obtain $C$ different predictions $P^1, ..., P^C$, we sample PriorNet $C$ times. We simultaneously feed these multiple predictions to the saliency consensus module to obtain the consensus of predictions.

Given multiple predictions $\{P^c\}_{c=1}^C$, where $P^c \in \{0, 1\}$, we first compute the binary version $P^c_b$ of the predictions by performing adaptive threshold [4] on $P^c$. For each pixel $(u, v)$, we obtain a $C$ dimensional feature vector $P_{u,v} \in \{0, 1\}$. We define $P^c_b(u,v) \in \{0, 1\}$ as a one-channel saliency map representing majority voting of $P_{u,v}$. We define an indicator $1^c(u,v) = 1(P^c_b(u,v) = P^c_{b mjv}(u,v))$ representing whether the binary prediction is consistent with the majority voting of the predictions. If $P^c_b(u,v) = P^c_{b mjv}(u,v)$, then $1^c(u,v) = 1$. Otherwise, $1^c(u,v) = 0$. We obtain one gray saliency map after saliency consensus as:

$$P^{mjv}_{g}(u,v) = \sum_{c=1}^{C} 1^{c}(u,v) \times P^{c}_{g}(u,v).$$

### 3.4. Objective Function

At this stage, our loss function is composed of two parts i.e., $L_{\text{CVAE}}$ and $L_{\text{Depth}}$. Furthermore, we propose to use the smoothness loss [9] as a regularizer to achieve edge-aware saliency detection, based on the assumption of inter-class distinction and intra-class similarity. Following [56], we define first-order derivatives of the saliency map in the smoothness term as

$$L_{\text{Smooth}} = \sum_{u,v} \sum_{d \in \vec{x}, \vec{y}} \Psi(|[\partial_d P_{u,v}] e^{-\alpha |[\partial_d I_g(u,v)]|}),$$

where $\Psi$ is defined as $\Psi(s) = \sqrt{s^2 + 1}e^{-s}$. $P_{u,v}$ is the predicted saliency map at position $(u, v)$, and $I_g(u,v)$ is the image intensity. $d$ indexes over partial derivative on $\vec{x}$ and $\vec{y}$ directions. We set $\alpha = 10$ following [56].

Both the smoothness loss (Eq. (5)) and the boundary IOU loss (Eq. (3)) need intensity $I_g$. We convert the RGB image $I$ to a gray-scale intensity image $I_g$ as [60]:

$$I_g = 0.2126 \times I_{r} + 0.7152 \times I_{g} + 0.0722 \times I_{b},$$

where $I_{r}$, $I_{g}$ and $I_{b}$ represent the color components in the linear color space after Gamma function being removed from the original color space. $I_{r}$ is achieved via:

$$I_{r} = \begin{cases} \frac{I'}{12.92}, & I' \leq 0.04045, \\ \left(\frac{I' + 0.055}{1.055}\right)^{2.4}, & I' > 0.04045. \end{cases} \quad (7)$$

where $I'$ is the original red channel of image $I$, and we compute $I_{g}$ and $I_{b}$ in the same way as Eq. (7).

As the GT map $Y \in \{0, 1\}$, we produce series of binary predictions with each one representing annotation from one saliency annotator.
With smoothness loss $L_{\text{Smooth}}$, depth loss $L_{\text{Depth}}$ and CVAE loss $L_{\text{CVAE}}$, our final loss function is defined as:

$$L_{\text{sal}} = L_{\text{CVAE}} + \lambda_1 L_{\text{Depth}} + \lambda_2 L_{\text{Smooth}}.$$  \hfill (8)

In our experiments, we set $\lambda_1 = \lambda_2 = 0.3$.

**Training details:** We set channel size of $S^d$ as $M = 32$, and scale of latent space as $K = 8$. We trained our model using Pytorch, and initialized the encoder of SalienceNet and DepthCorrectionNet with VGG16 parameters pre-trained on ImageNet. Weights of new layers were initialized with $\mathcal{N}(0, 0.01)$, and bias was set as constant. We used the Adam method with momentum 0.9 and decreased the learning rate 10% after each epoch. The base learning rate was initialized as 1e-4. The whole training took 13 hours with training batch size 6 and maximum epoch 30 on a PC with an NVIDIA GeForce RTX GPU. For input image size $352 \times 352$, the inference time is 0.06s on average.

### 4. Experimental Results

#### 4.1. Setup

**Datasets:** We perform experiments on six datasets including five widely used RGB-D saliency detection datasets (namely NJU2K [28], NLPR [41], SSB [40], LFSD [35], DES [8]) and one newly released dataset (SIP [18]).

**Competing Methods:** We compare our method with 18 algorithms, including ten handcrafted conventional methods and eight deep RGB-D saliency detection models.

**Evaluation Metrics:** Four evaluation metrics are used, including two widely used: 1) Mean Absolute Error (MAE $M$); 2) mean F-measure ($F_\beta$) and two recently proposed: 3) Enhanced alignment measure (mean E-measure, $E_C$) [15] and 4) Structure measure (S-measure, $S_o$) [14].

#### 4.2. Performance Comparison

**Quantitative Comparison:** We report performance of our method and competing methods in Table 1. It shows that our method consistently achieves the best performance on all datasets, especially on SSB [40] and SIP [18], our method achieves significant S-measure, E-measure, and F-measure performance boost and a decrease in MAE by a large margin. We show E-measure and F-measure curves of competing methods and ours in Fig. 7. We observe that our method produces not only stable E-measure and F-measure but also best performance.

**Qualitative Comparisons:** In Fig. 8, we show five images comparing results of our method with one newly released RGB-D saliency detection method (DMRA [61]), and two widely used methods to produce structured outputs, namely M-head [46] and MC-dropout [30] (we will discuss these two methods in detail in the ablation study section). We design both M-head and MC-dropout based structured saliency detection models by replacing CVAE with M-head and MC-dropout respectively. Results in Fig. 8 show that our method can not only produce high accuracy predictions (compared with DMRA [61]), but also diverse predictions (compared with M-head based and MC-dropout based models) for images with complex background (image in the first and last rows).

#### 4.3. Ablation Study

We carried out eight experiments (shown in Table 2) to thoroughly analyse our framework, including network structure (“M1”, “M2”, “M3”), probabilistic model selection (“M4”, “M5”, “M6”), data source selection (“M7”) and effectiveness of the new label generation technique (“M8”). We make the number bold when it’s better than ours.

**Scale of Latent Space:** We investigate the influence of the scale of the Gaussian latent space $K$ in our network. In this paper, after parameter tuning, we find $K = 8$ works best. We show performance with $K = 32$ as “M1”. Performance of “M1” is worse than our reported results, which indicates that scale of the latent space is an important parameter in our framework. We further carried out more experiments with $K \in [2, 12]$, and found relative stable predictions with $K \in [6, 10]$.

**Effect of DepthCorrectionNet:** To illustrate the effectiveness of the proposed DepthCorrectionNet, we remove this branch and feed the concatenation of the RGB image and depth data to the SaliencyNet, shown as “M2”, which is worse than our method. On DES [8] dataset, we observe the proposed solution achieves around 4% improvement on S-measure, E-measure and F-measure, which demonstrates the effectiveness of the depth correction net.

**Saliency Consensus Module:** To mimic the saliency labeling process, we embed a saliency consensus module during test in our framework (as shown in Fig. 3) to obtain the majority voting of the multiple predictions. We remove it from our framework and test the network performance by random sample from the latent PriorNet $P_\theta(z|X)$, and performance is shown in “M3”, which is the best compared with competing methods. While, with the saliency consensus module embedded, we achieve even better performance, which illustrates effectiveness of the saliency consensus module.

**VAE vs. CVAE:** We use CVAE to model labeling variants, and a PosteriorNet is used to estimate parameters for the PriorNet. To test how our model performs with prior of $z$ as a standard normal distribution, and the posterior of $z$ as $P_\theta(z|X)$. VAE performance is shown as “M4”, which is comparable with SOTA RGB-D models. With the CVAE [50] based model proposed, we further boost performance of “M4”, which proves effectiveness of our solution.

**Multi-head vs. CVAE:** Multi-head models [46] generate multiple predictions with different decoders and a shared encoder, and the loss function is always defined as the closest of the multiple predictions. We remove the LatentNet,
Table 1. Benchmarking results of ten leading handcrafted feature-based models and eight deep models on six RGBD saliency datasets. ↑ & ↓ denote larger and smaller is better, respectively. Here, we adopt mean $F_{0}$ and mean $E_{0}$ [15].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{ss}$</td>
<td>0.51</td>
<td>0.56</td>
<td>0.57</td>
<td>0.59</td>
<td>0.57</td>
<td>0.51</td>
</tr>
<tr>
<td>$F_{0}$</td>
<td>0.63</td>
<td>0.71</td>
<td>0.66</td>
<td>0.68</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>$E_{0}$</td>
<td>0.62</td>
<td>0.72</td>
<td>0.66</td>
<td>0.67</td>
<td>0.61</td>
<td>0.61</td>
</tr>
<tr>
<td>$M_{ss}$</td>
<td>0.43</td>
<td>0.56</td>
<td>0.57</td>
<td>0.58</td>
<td>0.57</td>
<td>0.51</td>
</tr>
<tr>
<td>$S_{ss}$</td>
<td>0.61</td>
<td>0.75</td>
<td>0.66</td>
<td>0.67</td>
<td>0.68</td>
<td>0.62</td>
</tr>
<tr>
<td>$F_{0}$</td>
<td>0.71</td>
<td>0.72</td>
<td>0.66</td>
<td>0.67</td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>$E_{0}$</td>
<td>0.72</td>
<td>0.72</td>
<td>0.67</td>
<td>0.67</td>
<td>0.61</td>
<td>0.61</td>
</tr>
<tr>
<td>$M_{ss}$</td>
<td>0.55</td>
<td>0.65</td>
<td>0.57</td>
<td>0.58</td>
<td>0.57</td>
<td>0.51</td>
</tr>
</tbody>
</table>

Deep Models

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{ss}$</td>
<td>0.73</td>
<td>0.78</td>
<td>0.73</td>
<td>0.73</td>
<td>0.72</td>
<td>0.75</td>
<td>0.72</td>
<td>0.74</td>
</tr>
<tr>
<td>$F_{0}$</td>
<td>0.82</td>
<td>0.87</td>
<td>0.83</td>
<td>0.81</td>
<td>0.80</td>
<td>0.85</td>
<td>0.83</td>
<td>0.84</td>
</tr>
<tr>
<td>$E_{0}$</td>
<td>0.83</td>
<td>0.87</td>
<td>0.83</td>
<td>0.81</td>
<td>0.80</td>
<td>0.85</td>
<td>0.83</td>
<td>0.85</td>
</tr>
<tr>
<td>$M_{ss}$</td>
<td>0.71</td>
<td>0.76</td>
<td>0.71</td>
<td>0.70</td>
<td>0.69</td>
<td>0.71</td>
<td>0.70</td>
<td>0.71</td>
</tr>
</tbody>
</table>

UC-Net

<table>
<thead>
<tr>
<th>Metric</th>
<th>NJU2K</th>
<th>SSB</th>
<th>DES</th>
<th>NLPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_{ss}$</td>
<td>0.63</td>
<td>0.63</td>
<td>0.62</td>
<td>0.62</td>
</tr>
<tr>
<td>$F_{0}$</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>$E_{0}$</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
<td>0.67</td>
</tr>
<tr>
<td>$M_{ss}$</td>
<td>0.59</td>
<td>0.59</td>
<td>0.59</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Figure 7. E-measure (1st row) and F-measure (2nd row) curves on four testing datasets.

and copy the decoder of the SaliencyNet multiple times to achieve multiple predictions ("M5" in this paper). We report performance in "M5" as mean of the multiple predictions. "M5" is better than SOTA models (e.g., DMRA) while there still exists gap between M-head based method ("M5") and our CVAE based model (UC-Net).

Monte-Carlo Dropout vs. CVAE: Monte-Carlo Dropout [30] uses dropout during the testing stage to introduce stochastic to the network. We follow [30] to remove the LatentNet, and use dropout in the encoder and decoder of the SaliencyNet in the testing stage. We repeats five times of random dropout (dropout ratio = 0.1), and report the mean performance as "M6". Similar to "M5", "M6" also achieves the best performance comparing with SOTA models (e.g., CPPF and DMRA), while the proposed CVAE based model achieves even better performance.

HHA vs. Depth: HHA [23] is a widely used technique that encodes the depth data to three channels: horizontal disparity, height above ground, and the angle the pixels local surface normal makes with the inferred gravity direction.
HHA is widely used in RGB-D related dense prediction models [11, 24] to obtain better feature representation. To test if HHA also works in our scenario, we replace depth with HHA, and performance is shown in “M7”. We observe performance improvement of “M9” compared with “M8”, which indicates effectiveness of the new label generation technique.

5. Conclusion

Inspired by human uncertainty in ground truth (GT) annotation, we proposed the first uncertainty network named UC-Net for RGB-D saliency detection based on a conditional variational autoencoder. Different from existing methods, which generally treat saliency detection as a point estimation problem, we propose to learn the distribution of saliency maps. Under our formulation, our model is able to generate multiple labels which have been discarded in the GT annotation generation process through saliency consensus. Quantitative and qualitative evaluations on six standard and challenging benchmark datasets demonstrated the superiority of our approach in learning the distribution of saliency maps. In the future, we would like to extend our approach to other saliency detection problems (e.g., VSOD [19], RGB SOD [13, 65], Co-SOD [17]). Furthermore, we plan to capture new datasets with multiple human annotations to further model the statistics of human uncertainty in interactive image segmentation [37], camouflaged object detection [16], etc.

Acknowledgments. This research was supported in part by Natural Science Foundation of China grants (61871325, 61420106007, 61671387), the Australia Research Council Centre of Excellence for Robotics Vision (CE140100016), and the National Key Research and Development Program of China under Grant 2018AAA0102803. We thank all reviewers and Area Chairs for their constructive comments.
References


