SaliencyGAN: Deep Learning Semi-supervised Salient Object Detection in the Fog of IoT

Chengjia Wang*, *Member, IEEE,* Shizhou Dong*, Xiaofeng Zhao, Giorgos Papanastasiou, Heye Zhang *Senior Member, IEEE,* and Guang Yang, *Member, IEEE,*

***Abstract*—In modern internet of things (IoT), visual analysis and predictions are often performed by deep learning models. Salient object detection (SOD) is a fundamental pre-processing for these applications. Executing SOD on the fog devices is a challenging task due to the diversity of data and fog devices. To adopt convolutional neural networks (CNN) on fog-cloud infrastructures for SOD-based applications, we introduce a semi-supervised adversarial learning method in this paper. The proposed model, named as SaliencyGAN, is empowered by a novel concatenated-GAN framework with partially shared parameters. The backbone CNN can be chosen flexibly based on the specific devices and applications. In the meanwhile, our method uses both the labelled and unlabelled data from different problem domains for training. Using multiple popular benchmark datasets, we compared state-of-the-art baseline methods to our SaliencyGAN obtained with 10% to 100% labelled training data. SaliencyGAN gained performance comparable to the supervised baselines when the percentage of labelled data reached 30%, and outperformed the weakly supervised and unsupervised baselines. Furthermore, our ablation study shows that SaliencyGAN were more more robust to the common "mode missing" (or "mode collapse") issue compared to the selected popular GAN models. The visualized ablation results proved that SaliencyGAN learned a better estimation of data distributions. To the best of our knowledge, this is the first IoT-oriented semi-supervised SOD method.**

***Index Terms*—Internet of Things, Deep Learning, Convolutional Neural Networks, Salient Object Detection, GAN.**

I. Introduction

EMpowered by 5G communication, the internet of things (IoT) collects and distributes large-scale streaming data from ubiquitous devices and objects [1]. A significant portion of these data are videos and images collected for a wide range of computer vision tasks. Salient object detection (SOD) is a fundamental pre-processing for these vision systems, for example, extracting suspicious events in smart home security camera. Recent developments in deep learning have have brought successful solution for many tasks in computer vision (CV), including SOD. However, the majority of these algorithms has not been customized to modern IoT systems. Specifically, due to the computing power, scale of the training data and diversity of CV applications required for training and utilizing deep neural networks, most SOD-based CV algorithms are deployed to the cloud infrastructure. These cloud devices are often equipped with high-performance graphic processing units to perform, for example, computation of large data matrices gathered from a vast variety of sensors. This will not only lead to longer response time but huge amount of manual effort for training the deep learning models. Our main motivation of this work is for a deep learning SOD methods that is flexible and efficient to be deployed on the cloud and fog devices of visual IoT.

SOD aims to extract visually distinctive contents from images and videos. It has been an explicit step for various computer vision applications in IoT [2]. Or SOD has been an implicit operation in deep CNN where a saliency map can be extracted from the hidden layers [3] (e.g., recognition, detection). These SOD-based computer vision techniques have been applied to a variety of IoT systems ranging from healthcare to industrial serveillance. SOD plays an important role in these IoT systems filtering out redundant background and facilitates fast post-processing [4]. As a result, SOD is more suitable to be deployed in the fog compared to the cloud devices, for example, smart home hub box and intelligent controller.

However, there are two main challenges of deploying modern SOD algorithms on fog devices. First, present SOD methods mostly use fully-supervised models, which requires manually generated pixel-wise ground truths. Although unsupervised [5] and weakly supervised methods have been proposed [6], they either have performance incomparable to supervised models or require impractically huge amount of image-wise annotations (details discussed in Section II).

Second, computing powers of the fog infrastructure are highly diverse, which leads to varying applicability of the same deep neural network. For example, a 19-layer VGGnet running smooth on a local GPU server may be not able to

∗Authors with equivalent contributions

Manuscript received July 23, 2019; revised September 4, 2019; accepted 25 September 2019. Date of publication XXX XXX, 2019; date of current version XXX XXX, 2019. Paper no. TII-19-2654.(Corresponding author:Heye Zhang. This work is supported in part by the Innovation funding of Guangdong Province (2018A050506031,2019B010110001), by the Fundamental Research Funds for the Central Universities, and by the National Natural Science Foundation of China (No: U1801265, 61771464).

Chengjia Wang is currently with the BHF Centre for Cardiovascular Science, The University of Edinburgh, Edinburgh EH16 4TJ, UK (e-mail: Chengjia.Wang@ed.ac.uk).

Shizhou Dong is currently with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: sz.dong@siat.ac.cn).

Xiaofeng Zhao is with School of Management Engineering and Business, Hebei University of Engineering, Handan 056038, China (e-mail: zhaoxiaofeng@hebeu.edu.cn).

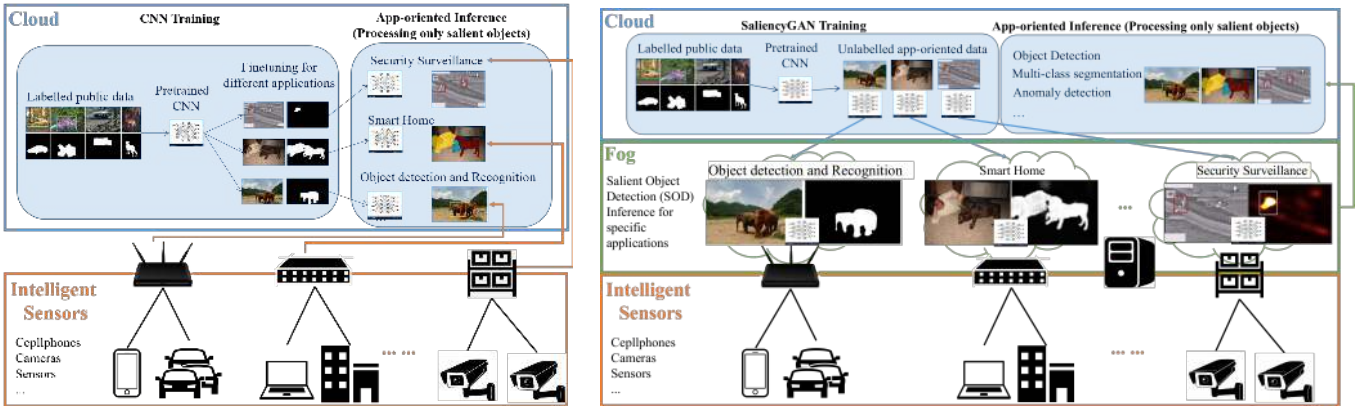
Giorgos Papanastasiou is with Edinburgh Imaging Facility QMRI, The University of Edinburgh, Edinburgh EH16 4TJ, UK (e-mail: g.papanas@ed.ac.uk).

Heye Zhang is with School of Biomedical Engineering, Sun Yat-Sen University, Guangzhou 510275, China (e-mail: heye.zhang@gmail.com).

Guang Yang is with Imperial College London, London SW7 2AZ, UK (e-mail: g.yang@imperial.ac.uk).

Color versions of one or more of the figures in this paper are availableonline at http://ieeexplore.ieee.org.

Digital Object Identifier XXXXXX



(a) cloud based SOD (b) fog based SOD

Fig. 1: Comparison of salient object detection (SOD) solutions in IoT: (a) present cloud based SOD; (b) the proposed SaliencyGAN method. In SaliencyGAN solution, pretrain of the model is performed on the cloud servers using public datasets. The backbone network was selected and finetuned based on specific applications and fog devices. In the test time, SOD are performed on the fog infrastructure as a general-purpose pre-processing. The filtered data with only salient contents are submitted to cloud for further computer vision tasks.

be executed on a cellphone. As a result, SOD models have mostly been implemented in the cloud servers together with the downstream post-processings, which harms the security of data privacy and bring difficulties for fast/real-time responses.

In this work, we propose an semi-supervised learning method, named as SaliencyGAN, for flexible and efficient deployment of different backbone CNNs in IoT systems. Using SaliencyGAN, we can first pretrain a deep SOD CNN in the cloud servers where this backbone CNN can be flexibly selected according to the computing power of fog devices. The pretrained model is then finetuned for a specific IoT application using both labelled and unlabelled data, and then distributed to different fog devices. In the test stage, these finetuned CNNs filter out background information and only submit the salient contents to the cloud servers for further post-processing. Compared to SaliencyGAN, current SOD solutions are mainly deployed on cloud. Figure 1 provides a comparison between present typical cloud-based solutions and our fog-based SaliencyGAN mode.

Figure 2 demonstrates fundamental idea behind our SaliencyGAN method: both the labelled and unlablled data are useful to estimate the distribution of the whole training dataset in a feature space, and this feature space has optimal separatibility for the background and salient regions; the salient objects can be extracted then through a mapping $f(\cdot)$ between the image feature space and a saliency map feature space. Training of the SOD model can be done through this joint learning process. To simulate this learning process, we use two concatenated generative adversarial networks (GAN) with partially shared parameters. The two GANs are trained end-to-end for simultaneous alignment of the labelled and unlabelled images within both the image feature space and saliency map feature space. In the meantime, the performance of both GANs are mutually reinforced. The image and saliency distributions shown in figure 2 were plotted from the popular MSRA10K [7] dataset using the two principle dimensions extracted by PCA.

The proposed SaliencyGAN model was assessed with multiple widely used benchmark datasets. Compared against selected fully-supervised baselines and state-of-the-art methods which use the same backbone CNN, SaliencyGAN achieved comparable performance with only 30% labelled images. We also argue that, compared to weakly supervised learning, the proposed method has better generalizability over real-world IoT applications. This is because of its capability transferring to new problem domains without categorical labelling. Furthermore, compared to present popular CycleGAN [8] and WGAN-GP [9] models in our simulation experiment, SaliencyGAN demonstrated better ability to capture the data distributions without sign of "mode missing".

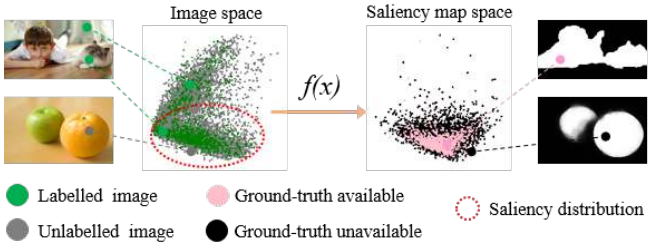


Fig. 2: Demonstration of our salient object detection (SOD) method using 10% labelled images from MSRA10K[7] dataset. Distributions of images and saliency maps (visualized using PCA) are estimated with an optimal mapping function. Both labelled and unlabelled images contribute to the final SOD performances without the common "mode missing" issue. (best viewed in color)

Contributions of this paper are summarized as follows:

1) We propose the semi-supervised SaliencyGAN adversarial learning framework that can achieve state-of-the-art performance with less ground truths. This enables fast and flexible deployment on a broad range of visual IoT systems. To the best of our knowledge, this is also the first semi-supervised method in the field of SOD.
2) We design the concatenated and mutually reinforced dual-GAN architecture, as well as the end-to-end traninig algorithm for simultaneous optimization of the unsupervised GAN losses with the supervised classifi-

cation loss. Any popular backbone CNN can be trained under this framework for specific fog/cloud devices and vision based tasks.

3) Comparison experiment results have demonstrated SaliencyGAN's capability of training any backbone DCNN with less labelled data, and capability of adapting a deep learning SOD model to new visual IoT tasks. Optimal hyperparametric setups have been verified by the ablation study. Its ability to capture multimodal data distribution without the common "mode missing" problem has been proved by our simulation experiment.

The paper is organized as follows: Section II reviews the current SOD models and related semi-supervised learning methods. Section III gives the details of the SaliencyGAN method. SectionIV presents details about implementation and experiments. Experiment results are analyzed and discussed Section V, and section VI concludes this paper.

II. Related work

Deep CNNs have been applied to a broad range of IoT systems, such as, material retrieval [10], security surveillance [11], and pipeline anormaly inspection [12]. SOD is performed explicitly in [11] where the salient maps indicate objects moving between video frames. For IoT systems proposed in [12] SOD is an implicit step performed in the hidden layers where a saliency map can be extracted. Compared to early SOD models using handcrafted features that describe low-level image properties, such as color, edge, intensity, texture and local contrast, these deep learning models mostly look at learning rich feature representations through hierarchical and multi-stage refinement, or using ingenious network architecture that can obtain more effective features [13]. Because previous works focus on downstream computer vision techniques in particular IoT systems, performance and applicability of the involved SOD algorithms have not been evaluated. Furthermore, the SOD prediction step of these models are deployed on the cloud as a pre-processing. Efforts for an independent fog-oriented SOD model is missing. In the field of computer vision, recent SOD models which achieved state-of-the-art performance [14][15] are most fully supervised learning as in the discussed visual IoT applications. This requires sufficient amount of pixel-level labels that are expensive to obtain. Unsupervised SOD [5] models allows to train solely unlabelled data, but the reported performances are still hard to be comparable to supervised state-of-the-art methods.

Recently, a weakly-supervised SOD model [6], which is trained with only image-level categorical labels, has shown superior performance compared to unsupervised methods. However, the huge amount of training images use for training the weakly-supervised method can still lead to time-consuming manual labelling process. Specifically, in [6], 456k images were used for training the weakly supervised model while most supervised methods were trained on benchmark datasets with 10,000 images. According a recent study [16] which provides an estimation of time used for generating image-level labels ($\sim$1 second/image) and for generating ground-truth masks of objects (40 second/image), preparing training data for weakly-supervised SOD models can be potentially more label-intensive. Furthermore, as a general-purpose preprocessing, the data processed by SOD methods should not be limited to certain categories. For a broad range of IoT application (see figure 1), it is also impractical to quickly adopt these models to new problem domains.

Adversarial learning [17] has been found helpful to SOD models for better generalizability and robustness [18]. For modern IoT applications where massive images and videos can be easily acquired, GAN-based models are useful for adjusting the feature spaces established by deep neural networks to the quickly changed data variance. For instance, Zheng et al.[10] developed a visual-tactile cross-modal retrieval framework surface material retrieval. Futhermore, adversarial learning is also useful to secure data privacy and reduce security cost in IoT. This process has often been implemented as a collaborative learning through adversarial attack [19]. However, a common issue in all GAN-based deep learning models is "mode missing". A popular method to improvement the convergence of the original GAN method is to replacing the cross-entropy loss with the Wasserstein distance and using gradient penalty [9]. Zhu et al. [8] introduced the cycle-consistency loss function for better estimation of data distributions.

III. Method

A. SaliencyGAN Architecture

As introduced above, we train a SOD CNN within a concatenated-GAN framework deployed on the cloud, and adopt the trained CNN to different fog devices in IoT. Figure 3 displays the SaliencyGAN architecture. The proposed model consists of two concatenated GAN frameworks, each has a generator and a discriminator network. The first GAN includes an image generator G^I which generates fake images from a input random latent code, and an image discriminator D^I which is trained to distinguish real images from the training dataset and fake images generated by G^I. In the mean time, G^I is trained to confuse D^I. The second GAN consists of a saliency generator G^S that performs SOD prediction and a saliency discriminator D^S. D^S is trained to predict whether the saliency maps are extracted from the unlabelled images or from the labelled images. G^S on one hand is trained to predict accurate saliency maps for the labelled data, and to confuse D^S on the other hand. For real applications in IoT, the labelled data can be obtained from public datasets, and data acquired for a specific IoT system can be unlabelled and directly used for training in cloud. After training finished, only G^S need to be distributed to the fog for SOD prediction.

The general architecture of SaliencyGAN evaluated in this work follow the design of DCGAN [20]. Each of the two discriminator networks, D^I and D^S, has four 4×4 convolutional layers with stride 2, followed by a global average pooling (GAP) layer and a fully-connected layer. The ith convolutional layer in D^I has $128 \cdot i$ channels. The number of channels in corresponding layers of D^S only is only $1/8$ of D^I, as distinguishing saliency maps is intuitively easier than distinguishing images. The final predictions are single scalar values output from a *Sigmoid* layer. To perform SOD

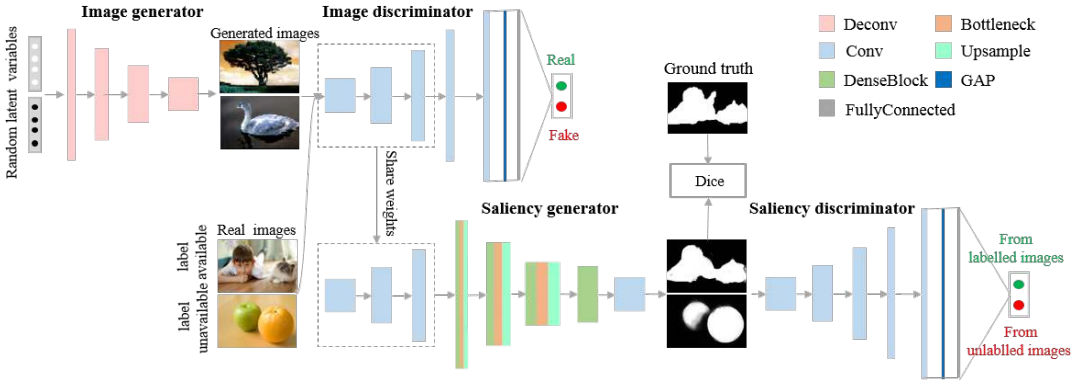


Fig. 3: The proposed SaliencyGAN framework. The global average pooling layer (GAP) is placed at the end of image and saliency discriminator. For a real IoT system, only the saliency generator which consists of a set of"Denseblock" and "Bottleneck" layers in this example is distributed to the fog devices. (best viewed in color)

in the fog, architecture of G^S can be selected according to different applications. Regardless of the backbone network, the first three layers of G^S share weights with D^I. In this work, we evaluated two backbone networks in our experiments. For comparison with state-of-the-art baselines, we use VGG16 structure. Another tested backbone network is shown in figure 3, where DenseNet and Resnet Bottleneck blocks are stacked after the first three convolutional layers.

The concatenated GAN structure aims to precisely model distributions of the image and saliency-map in the feature spaces. In the first GAN, training of G^I forces D^I to encode both the labelled and unlabelled images from datasets into a uniform distribution. Similarly, to confuse D^S, G^S must harmoniously combine features from the labelled and unlabelled data, and obtain indistinguishable performance on both. As a result, all input data contributes to the training of SOD. The shared weights between D^I and G^S play a critical role in SaliencyGAN. This design forces D^I to use saliency features in the input images for its final prediction.

B. The SaliencyGAN loss and training pipeline

To sum up, the SaliencyGAN framework consists of four subnetworks: the image generator G^I, the image discriminator D^I, the saliency generator G^S and the saliency discriminator D^S. These networks are jointly trained using minibatch gradient descent. For each of the four subnetworks we define an adversarial loss. For G^S, an extra supervised loss is defined between the output of G^S and the labelled groud truths.

1) G^I and D^I Adversarial Loss: In D^I and D^S, a GAP layer transform the feature maps in 1D feature vectors before input into the fully connected layers. Let $D^I_{GAP}(*)$ denote the features output by the GAP layer of D^I. To capture the feature distribution of images, we use a feature matching loss, $\mathcal{L}^I_{fea}$, originally proposed in [21] for G^I:

$$\mathcal{L}^I_{fea}(\hat{x}, x) = \left[\mathop{\mathbb{E}}_{\hat{x}\sim P_{fi}} D^I_{GAP}(\hat{x}) - \mathop{\mathbb{E}}_{x\sim P_{ri}} D^I_{GAP}(x) \right]^2, \quad (1)$$

where x and $\hat{x}$ are a real image and a fake image generated by G^I, and P_{ri} and P_{fi} are the feature distributions of real images and fake images. The adversarial loss of the image generator G^I is then defined by:

$$\mathcal{L}_{G^I,adv}(\hat{x}, x) = - \mathop{\mathbb{E}}_{\substack{x\sim P_{ri} \\ \hat{x}\sim P_{fi}}} \left[D^I(\hat{x}) \right] + \mathcal{L}^I_{fea}(\hat{x}, x). \quad (2)$$

For D^I, we use the adversarial loss based on Wasserstein distance and gradient penalty in WGAN-GP [9]:

$$\begin{aligned} \mathcal{L}_{D^I,adv}(\hat{x}, x) = & \mathop{\mathbb{E}}_{\substack{x\sim P_{ri} \\ \hat{x}\sim P_{fi}}} \left[-D^I(x) + D^I(\hat{x}) \right] + \\ & \lambda_I \mathop{\mathbb{E}}_{\tilde{x}\sim P_{penalty}} \left[\left(\left\| \nabla_{\tilde{x}} D^I(\tilde{x}) \right\| - 1 \right)^2 \right], \end{aligned} \quad (3)$$

where the gradient panelty weight $\lambda_I = 10$. This loss has been proved to have better stability of adversarial training.

2) G^S and D^S Adversarial Loss: Similarly, for the saliency discriminator D^S, the feature matching loss $\mathcal{L}^S_{fea}$ is defined:

$$\begin{aligned} & \mathcal{L}^S_{fea}(\hat{s}_l, \hat{s}_u) = \\ & \left[\mathop{\mathbb{E}}_{\hat{s}_l\sim P_{ls}} D^S_{GAP}(\hat{s}_l) - \mathop{\mathbb{E}}_{\hat{s}_u\sim P_{us}} D^S_{GAP}(\hat{s}_u) \right]^2, \end{aligned} \quad (4)$$

where $\hat{s}_l$ and $\hat{s}_u$ are the saliency maps of labelled and unlabelled input images, generated by G^S. Let P_{ls} and P_{us} represent the probability distributions of saliency maps generated from labelled and unlabelled data, and $D^S_{GAP}(*)$ represents the features output by the GAP layer of D^S. Induction of $\mathcal{L}^S_{fea}$ reduces statistics differences between saliency maps generated from labelled and unlabelled images, thus allows to effectively merge the distributions of labelled and unlabelled within the feature spaces of D^S. As D^S has much less trainable parameters compared to D^I, we found the binary entropy loss without gradient penalty is sufficient for a stable convergence. The adversarial losses of G^S and D^S are then defined as:

$$\begin{aligned} \mathcal{L}_{G^S,adv}(\hat{s}_l, \hat{s}_u) = & \mathop{\mathbb{E}}_{\hat{s}_u\sim P_{us}} log \left[1 - D^S(\hat{s}_u) \right] \\ & + \mathcal{L}^S_{fea}(\hat{s}_l, \hat{s}_u), \end{aligned} \quad (5)$$

$$\begin{aligned} \mathcal{L}_{D^S,adv}(\hat{s}_l, \hat{s}_u) = & - \mathop{\mathbb{E}}_{\hat{s}_l\sim P_{ls}} log \left[D^S(\hat{s}_l) \right] \\ & - \mathop{\mathbb{E}}_{\hat{s}_u\sim P_{us}} log \left[1 - D^S(\hat{s}_u) \right]. \end{aligned} \quad (6)$$

3) G^S Supervised Loss: The adversarial losses can be viewed as unsupervised losses as they solely use the predicted saliency maps of G^S rather than the ground-truth labels. G^S, as the network finally used for SOD, can be trained a supervised loss. Because the SOD datasets suffer from severe label imballance, here we use the *Generalised Dice Loss* (GDL) proposed in [22]:

$$\mathcal{L}_{G^S,sup} = GDL(\hat{s}_l, s_l), \tag{7}$$

where s_l and $\hat{s}_l$ are the real and predicted saliency maps of a labelled image. GDL is an extension of binary Dice loss where the foreground and background are reweighed based on numbers of pixels. It has been widely used for segmentation to deal with unbalanced class labelling.

Algorithm 1 Minibatch training of SaliencyGAN.

Require:

1: Learning rate η.
2: Prior distribution $P_{prior}(z)$.
3: Weight of saliency generator adversarial loss ω.

Ensure:

4: **for** each iteration in training **do**
5: **Sample** m examples $\{x_l^1, x_l^2, ...x_l^m\}$ from the labelled data. Their ground truth are $\{s_l^1, s_l^2, ...s_l^m\}$, m examples $\{x_u^1, x_u^2, ...x_u^m\}$ from unlabelled images, and merge the $2m$ images to $\{x^1, x^2, ...x^{2m}\}$.
6: **Sample** $2m$ noise examples $\{z^1, z^2, ...z^{2m}\}$ from $P_{prior}(z)$.
7: **Generate** $2m$ image samples $\{\hat{x}^1, \hat{x}^2, ...\hat{x}^{2m}\}$ using the G^I, by $\hat{x}^i = G^I(z^i)$, and m predicted saliency maps for labelled and unlabelled images $\{\hat{s_l}^1, \hat{s_l}^2, ...\hat{s_l}^m\}$ and $\{\hat{s_u}^1, \hat{s_u}^2, ...\hat{s_u}^m\}$ separately by $\hat{s}^i = G^S(x^i)$ and and combine the $2m$ saliency maps into $\{\hat{s}^1, \hat{s}^2, ...\hat{s}^{2m}\}$.
8: **Update** the parameters θ_{D^S} of D^S by:

$$\theta_{D^S} \leftarrow \theta_{D^S} - \eta \times \bigtriangledown_{\theta_{D^S}} \left[\frac{1}{m} \sum_i^m \mathcal{L}_{D^S,adv}(\hat{s_l}^i, \hat{s_u}^i) \right].$$

9: **Update** the parameters θ_{D^I,G^S} of D^I and G^S by:

$$\begin{aligned}\theta_{D^I,G^S} \leftarrow & \theta_{D^I,G^S} \\ & - \eta \times \bigtriangledown_{\theta_{D^I,G^S}} \left[\frac{1}{2m} \sum_i^{2m} \mathcal{L}_{D^I,adv}(x^i, \hat{x}^i) \right. \\ & + \frac{1}{m} \sum_i^m \mathcal{L}_{G^S,sup}(s_l^i, \hat{s}_l^i) \\ & \left. + \frac{\omega}{m} \sum_i^m \mathcal{L}_{G^S,adv}(\hat{s}_u^i, \hat{s}_l^i) \right]\end{aligned}$$

.
10: **Update** the parameters θ_{G^I} of Image Generator G^I.

$$\theta_{ig} \leftarrow \theta_{ig} - \eta \times \bigtriangledown_{\theta_{ig}} \left[\frac{1}{2m} \sum_i^{2m} \mathcal{L}_{G^I,adv}(\hat{x}^i, x^i) \right]$$

.
11: **end for**

C. Training SaliencyGAN

Based on the architecture of SaliencyGAN, a training pipeline is designed for step-wise update of the supervised loss $\mathcal{L}_{G^S,sup}$, and the adversarial losses $\mathcal{L}_{G^I,adv}$, $\mathcal{L}_{D^I,adv}$, $\mathcal{L}_{G^S,adv}$ and $\mathcal{L}_{D^S,adv}$. Algorithm 1 illustrates the training process of SaliencyGAN. In each iteration, the model takes m labelled images and m unlabelled images as input, as well as $2m$ fake images generated from prior distribution $P_{prior}(z)$. In backpropagation, parameters of D^S, (denoted by θ_{D^S} in Algorithm 1), are updated first using $\mathcal{L}_{D^S,adv}$, followed by θ_{D^I,G^S} (parameters of D^I and G^S which are partially shared) using $\mathcal{L}_{D^I,adv}$, $\mathcal{L}_{G^S,sup}$ and $\mathcal{L}_{G^S,adv}$. As shown in the 9th step of Algorithm 1, a weight, ω, is used to balance the supervised loss and adversarial losses. The value of ω is set to 10^{-4} based on a grid search. Parameters of G^I are updated at last.

IV. Experiments

A. Experiment Design

We comprehensively evaluated the performances of SaliencyGAN as a semi-supervised adversarial SOD model in three comparison experiments and two ablation study.

1) Comparison Experiments: First, precisely capturing data distribution is critical to the performance of SaliencyGAN. But "mode missing" problem is a common issue in adversarial learning methods. So we first compared SaliencyGAN with popular GAN-based models in a task of modeling multi module Gaussian distributions in *Comparison Experiment 1*. The fundamental idea behind all deep learning SOD model is to train a CNN which predicts a saliency map from a input image. In SaliencyGAN, this CNN is the saliency generator G^S. G^S also can be trained in fully-supervised or adversarial approaches with all available labelled data. In the second experiment, we use these two approaches as baselines. We compared performance of different versions of G^S trained by supervised and adversarial approaches, as well as the proposed semi-supervised approach in *Comparison Experiment 2*. As a semi-supervised method, different combinations of labelled and unlabelled data can be used to train the SaliencyGAN. Let N_l and N_u be the numbers of labelled and unlabelled images in a dataset, we define the labelling ratio $p_l = \frac{N_l}{N_l+N_u} \cdot 100\%$. We gradually increase p_l to find a minimum ratio of labelled data where SaliencyGAN can obtain comparable performance with the supervised baselines. This combination of labelled and unlabelled data is used for further comparison with state-of-the-art methods in the third Experiment (*Comparison Experiment 3*).

In Comparison Experiment 3, VGG16 architecture was adopted for G^S, and in other comparison experiments we used the structure shown in figure 3.

2) Ablation Studies: We performed two ablation experiments to evaluate the hyperparametric configuration and the proposed training pipeline. The weight ω of saliency generator adversarial loss $\mathcal{L}_{G^S,adv}$ shown in Algorithm 1 step 9 has a significant effect in backpropagation. The first ablation study (Ablation Experiment 1) looks at the influence of ω on the final SOD performances. Furthermore, in the proposed training Algorithm 1, the two sets of GAN are updated simultaneously

but they can also converge when trained separately. The second ablation study (*Ablation Experiment 2*) validate the importance of this end-to-end training procedure compared to separately training the two sets of GANs. Again we tested two backbone structures for G^S: the achitecture shown in figure 3 for Ablation Study 1 and VGG16 for Ablation Study 2.

B. Datasets

In Comparison Experiment 1, we use 1D simulated data. Other experiments were performed using public benchmark SOD datasets. To be consistent with the prior works, we train the compared methods on MSRA10K [7]. In total six popular benchmark datasets were used for testing: DUT-OMRON (5148 images) [23], DUT-TE (5019 images) [24], ECSSD (1000 images) [25], HKU-IS (4447 images) [26], PASCAL-S (850 images) [27], THUR-15K (6233 images) [28]. These datasets contain a wide range of natural images collected from indoor and outdoor environments, which provide abundant training materials for IoT applications, such as, smart cities and smart home. All the six benchmark datasets were used in Comparison Experiment 2 and Ablation Experiment 1. For Comparison Experiment 3 and Ablation Experiment 2, we compared performance on the DUT-TE, ECSSD, HKU-IS, PASCAL-S, THUR-15K datasets.

C. Implementation Details

To simulate a IoT environment, we train and test a SOD CNN using different GPUs. Training of SaliencyGAN were performed on a server equipped with 4 nVidia Tesla P100 GPUs which simulates the computing units in a data center. The trained G^S is then tested on a smaller Tesla K80 GPU with 12G memory. To simulate a typical fog device, we limited the available memory of the K80 GPU to 4 gigabytes. This is smaller than the GPU memory on present personal PCs which are typically used as a fog device. We implement SaliencyGAN in Python in TensorFlow. All the models are trained by Adam optimizer with β_1 0.5 and β_2 0.999. The initial learning rate η is set to 0.0002. The batch size m is 16 and the unsupervised loss weight ω is 0.0001. The prior distribution $P_{prior}(z)$ is $N(0, I)$ normal distribution.

D. Evaluation metres

Results of Ablation Experiment 1 was assessed qualitatively. In other experiments, four metrics were used to quantitatively measure the performances of models: pixel-wise mean absolute error (MAE) of saliency maps, maximum F-measure scores (maxF), Precision Recall curves (PR-curves), F-measure curves to quantitatively evaluate the experiment results. The maxF is the max value of F-measure scores among 100 discrete thresholds in the range [0,1], which reflects an overall performance between S and G. The F-measure is defined by $F_t = \frac{(1+0.3)\cdot Precision_t \cdot Recall_t}{0.3\cdot Precision_t + Recall_t}$ as in [13], where $Precision_t$ and $Recall_t$ are obtained by using a threshold t. Moreover, the PR-curve is computed as the mean precision and recall values at different thresholds. The F-measure curve is computed as the F-measure scores at different thresholds. Except when being compared to state-of-the-art models, we use a saliency generator G^S with the light-weight architecture shown in figure 3 for better applicability to fog IoT devices.

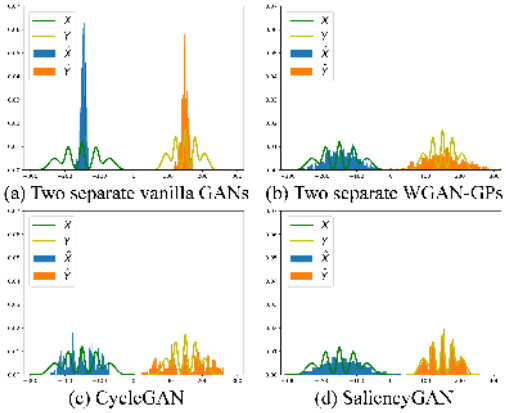


Fig. 4: Visualization of distributions estimated by SaliencyGAN, as well as by vanilla GAN, CycleGAN and WGAN-GP on capturing data distributions. X and Y mean input and output data separately. $\hat{X}$ and $\hat{Y}$ mean generated input and output data by models. (best view in color and with zoom)

V. Results and Discussion

A. Comparison 1: SaliencyGAN vs. modern GAN-based models on capturing data distributions

The first experiment uses simulated data to assess the capability of avoiding the "mode missing" issue. The training dataset was generated by sampling two multimodal Gaussian distributions which simulate the distributions of input images X and output saliency-maps Y. A mapping function were learned while the two distributions were estimated by GAN-based models. Estimated distributions obtained by SaliencyGAN and three baseline GAN-based models (vanilla GAN [17], CycleGAN [8] and WGAN-GP [9]) were plotted for comparison. This experiment provides a insight into SaliencyGAN's ability to accurately capture data distributions.

Figure 4 shows the distributions estimated by different GAN-based models, where X and Y represent the ground-truth distributions of input and output, and $\hat{X}$ and $\hat{Y}$ are the predicted distributions. As shown in Figure 4, because vanilla GAN is only trained to converge to samples with highest discriminator loss, the generator collapses which produces limited varieties of samples. CycleGAN uses cycle-consistent loss to make the mapping $\hat{X} \rightarrow \hat{Y}$ revertible. This reduces the effects of "mode missing", but still has no explicit mechanism to keep the varieties of samples. WGAN-GP improved estimation of data distribution by using Wasserstein distance to minimize the difference of two distributions. However, precise capturing the data distributions requires large amount of samples. Otherwise the estimated distributions will be noisy. SaliencyGAN gave the most accurate estimation of output data distributions because of the shared weights between D^I and G^S, which introduced classification error (classification of saliency and background) into the GAN loss and preserve the

Label (%)	Method	DUT-OMRON[23]		DUT-TE[24]		ECSSD[25]		HKU-IS[26]		PASCAL-S[27]		THUR-15K[28]	
		MAE ↓	maxF ↑	MAE ↓	maxF ↑	MAE ↓	maxF ↑	MAE ↓	maxF ↑	MAE ↓	maxF ↑	MAE ↓	maxF ↑
10%	G^S	0.1406	0.5879	0.1500	0.5680	0.1327	0.7389	0.1156	0.7191	0.1722	0.6751	0.1430	0.5960
	G^S+D^S	0.1336	0.6018	0.1435	0.5845	0.1290	0.7465	0.1143	0.7286	0.1740	0.6740	0.1409	0.6067
	G^I+D^I+G^S+D^S	**0.1306**	**0.6102**	**0.1349**	**0.6100**	**0.1156**	**0.7763**	**0.1071**	**0.7418**	**0.1637**	**0.6990**	**0.1322**	**0.6217**
30%	G^S	0.1377	0.5953	0.1412	0.5959	0.1172	0.7698	0.1060	0.7465	0.1656	0.6932	0.1355	0.6219
	G^S+D^S	0.1325	0.6065	0.1386	0.6033	0.1153	0.7776	0.1090	0.7504	0.1652	0.6950	0.1324	0.6267
	G^I+D^I+G^S+D^S	**0.1126**	**0.6479**	**0.1245**	**0.6288**	**0.1004**	**0.8070**	**0.0928**	**0.7759**	**0.1512**	**0.7228**	**0.1225**	**0.6416**
50%	G^S	0.1242	0.6217	0.1340	0.6120	0.1107	0.7887	0.0977	0.7656	0.1549	0.7153	0.1296	0.6328
	G^S+D^S	0.1098	0.6515	0.1209	0.6358	0.1059	0.7969	0.0900	0.7848	0.1541	0.7172	0.1194	0.6473
	G^I+D^I+G^S+D^S	**0.1131**	**0.6545**	**0.1226**	**0.6425**	**0.0953**	**0.8176**	**0.0883**	**0.7885**	**0.1467**	**0.7295**	**0.1204**	**0.6556**
70%	G^S	0.1259	0.6192	0.1312	0.6200	0.1066	0.7949	0.0965	0.7689	0.1574	0.07132	0.1277	0.6376
	G^S+D^S	0.1156	0.6407	0.1259	0.6303	0.1024	0.8051	0.0926	0.7766	0.1530	0.7228	0.1246	0.6402
	G^I+D^I+G^S+D^S	**0.1121**	**0.6527**	**0.1225**	**0.6415**	**0.0950**	**0.8167**	**0.0868**	**0.7880**	**0.1447**	**0.7364**	**0.1170**	**0.6540**
90%	G^S	0.1208	0.6317	0.1270	0.6269	0.1036	0.7989	0.0931	0.7728	0.1518	0.7197	0.1266	0.6393
	G^S+D^S	0.1209	0.6372	0.1271	0.6335	0.1039	0.8017	0.0922	0.7789	0.1527	0.7239	0.1221	0.6476
	G^I+D^I+G^S+D^S	**0.1068**	**0.6680**	**0.1163**	**0.6545**	**0.0893**	**0.8307**	**0.0826**	**0.8005**	**0.1437**	**0.7385**	**0.1140**	**0.6623**
100%	G^S	**0.1179**	**0.6326**	**0.1258**	**0.6276**	**0.1069**	**0.7979**	**0.0931**	**0.7776**	**0.1554**	**0.7353**	**0.1212**	**0.6443**

TABLE I: MAE and maxF results of comparing SaliencyGAN with supervised and adversarial baselines on six benchmark datasets using the same backbone network. "G^S" represents supervised baseline where the image generator G^S is trained solely with the supervised loss; "$G^S + D^S$" represents adversarial baseline where "G^S" is trained with supervised and saliency adversarial losses; the complete SaliencyGAN method is represented as 'G^I+D^I+G^S+D^S'. All the tested models are trained on the MSRA10K dataset where the proportion of labelled data varies between 10% and 100%. Lower MAE and higher maxF indicate better performances.

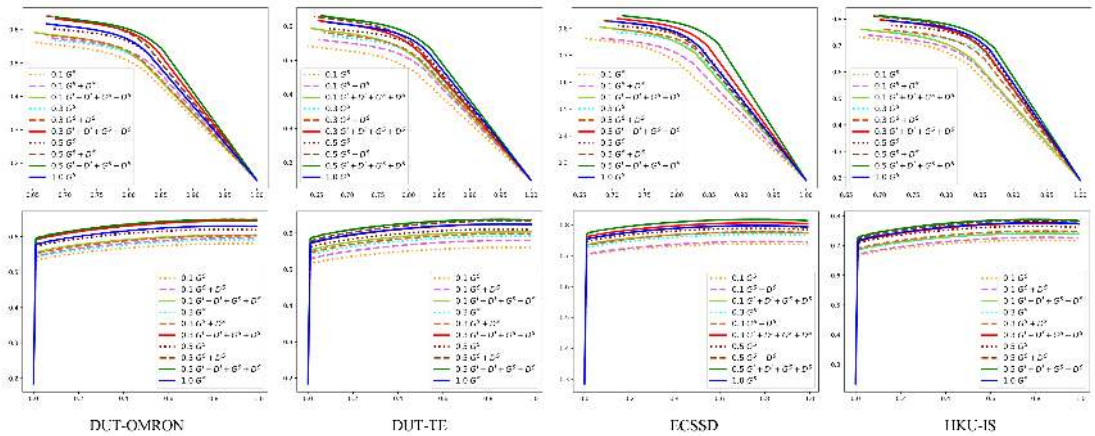


Fig. 5: PR curves and F-measure curves of four benchmark datasets. The first line is PR curves, where the horizontal and vertical axes are recall and precision respectively. The second line is F-measure curves, where the horizontal and vertical axes are threshold and F-measure score. (best viewed in color and with zoom)

variance in the estimated data distribution. Furthermore, semi-supervised training enables G^S to preserve larger uncertainty in distributions of each subclass.

B. Comparison 2: SaliencyGAN vs. ablated fully supervised benchmarks

As explained above, we trained the saliency generator G^S with fully supervised loss and trained G^S and D^S with adversarial loss, both using labelled data in MSRA10K. This gave us a supervised baseline and an adversarial baseline. Performance of SaliencyGAN method trained from scratch on MSRA10K were them compared to the two baseline. To estimate the optimal combination of labelled and unlabelled data, we collected results percentages labelled data ($p_l = 10\%, 30\%, 50\%, 70\%, 90\%, 100\%$).

Table I presents the quantitative results of the compared models obtained on the six benchmark datasets, where MAE and maxF were used as the metrics of performances. For simplicity, we use the involved subnetworks to represent the compared methods: "G^S" represents the supervised baseline, "G^S + D^S" the adversarial baseline, and "G^I + D^I + G^S + D^S" the full SaliencyGAN framework. It can be seen

that, performance of the two baselines increased significantly when p_l growing from 10% to 90%. For SaliencyGAN, when p_l is larger than 30%, it obtained comparable performance with supervised baseline trained with 100% ground truths. Further increment of p_l didn't lead to obvious improvement of SaliencyGAN's performance. The adversarial baseline, "G^S+D^S", gave better SOD results than the supervised baseline "G^S". In most cases, the SaliencyGAN model outperforms the "G^S+D^S" model. Figure 5 presents the PR-curves and F-measure curves of the compared methods. It shows that SaliencyGAN trained with $p_l = 50\%$ gave higher PR-curves and F-measure curves than the supervised baseline trained with 100% labelled images. When $p_l = 30\%$, SaliencyGAN achieved comparable performances with the fully supervised and adversarial baselines. Increasing p_l slightly improves the performances of all models, but less obvious for SaliencyGAN. For convenient visual inspection, some examples of saliency maps produced by different methods are shown in Figure 6.

C. Comparison 3: Comparison with state-of-the-art methods

We compared performances of SaliencyGAN and state-of-the-art models on five benchmark datasets. Unsupervised methods (MST [29], DRFI [5]) were used as baselines for a lower bound of performance. Three popular fully supervised methods, DCL+ [14], SRM [15], and HDFP [13] were selected as upper bond baselines. Most importantly, we also compared SaliencyGAN the weakly supervised model WSS [6]. Because $p_l = 30\%$ is the minimum ratio of labelled images for SaliencyGAN to obtain performance comparable to fully-supervised and adversarial baselines, we use only 30% of ground-truth saliency maps to train SaliencyGAN. The size of the input images were set to 128×128 and 256×256. Quantitative results are shown in table II. Results of the best performed method on each dataset are presented in red. Results of the SaliencyGAN methods that outperformed the weakly supervised model are shown in bold.

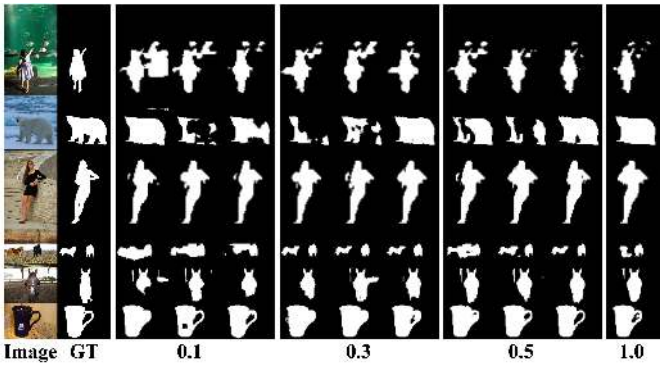


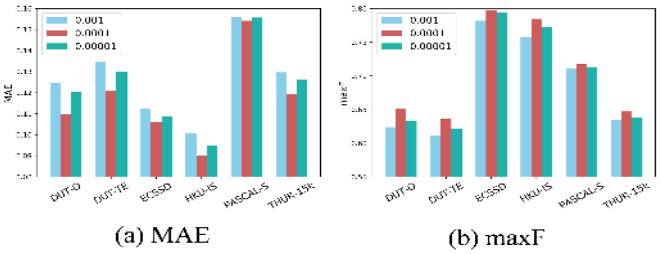
Fig. 6: Visualized examples of saliency maps generated the SaliencyGAN model under different configurations. From top to bottom, the input images are from the dataset of DUT-OMRON, DUT-TE, ECSSD, HKU-IS, PASCAL-S and THUR-15K. From left to right, each column shows the results obtained when the ratio of labelled data is 10%, 30%, 50% and 100%. In each column, the three subcolumns are the results of supervised baseline ("G^S"), the adversarial baseline ("G^S+D^S") and the proposed SaliencyGAN ("G^I + D^I + G^S + D^S") from left to right.

Based on the MAE and F-measure values, our SaliencyGAN outperformed the WSS on 4 out of 5 datasets, and displayed comparable performance with fully supervised state-of-the-art models. The only exception is the result obtained on the THUR-15K dataset where WSS even achieved lower MAE value than the fully supervised methods. This may because THUR-15K provides categorical data from only 5 classes. These classess are all overlapped with the ImageNet data with which WSS was pretrained.

Note that the unsupervised models are initiated by VGG weights which are trained on ImageNet (1281k images with over 1000 category labels), the weakly-supervised method is trained on a ImageNet subset of 456k images with over 200 category labels. Our SaliencyGAN was trained from scratch with only 3k ground-truth saliency maps. According to [16], manual annotation for training SaliencyGAN requires less than 1/3 labelling time of training WSS.

D. Ablation Study 1: Influence of ω

We validated the chosen value of the weight ω (see Algorithm 1 step 9) using all the six test datasets. Figure 7 shows the performance of SaliencyGAN when different ω (0.001, 0.0001, 0.00001) is assigned to the saliency generator adversarial loss $\mathcal{L}_{G^S,adv}$. When $\omega = 0.0001$, SaliencyGAN achieves the best performance. Too small ω might lead to a drop of SOD accuracy. Too large ω might force the model to focus on generating a saliency map that looks real, but ignores the input image.



(a) MAE (b) maxF

Fig. 7: Comparison of SaliencyGAN with different ω assigned to the saliency generator adversarial loss. Quantitative results are obtained from the "$G^S + D^S$" model trained with 50% labelled data. "DUT-O" represents the DUT-OMRON dataset. (best view in color)

E. Ablation 2: End-to-end v.s. separate training

To validate the importance of our end-to-end training procedure (joint optimization shown in Algorithm 1), we also separately trained the two sets of GANs in the proposed model with the datasets used in Comparison Experiment 3. We first trained G^I and D^I until converge, then G^S and D^S. All images are scaled to 128×128. Table III show the performances of G^S trained by these two methods. Separating the training process of the two sets of GANs led to worse performance based on both MAE and F-measure results. This ablated model only slightly outperformed the unsupervised baseline methods.

Dataset	Metric	Unsupervised		Weakly	Supervised			SaliencyGAN (30% labelled)	
		MST[29]	DRFI [5]	WSS [6]	DCL+ [14]	SRM [15]	HDFP [13]	SaliencyGAN-128	SaliencyGAN-256
DUT-TE [23]	MAE ↓	0.163	0.155	0.100	0.082	0.059	0.061	**0.095**	**0.090**
	F-measure ↑	0.594	0.650	0.737	0.786	0.826	0.827	0.730	**0.748**
ECSSD [25]	MAE ↓	0.157	0.170	0.104	0.068	0.054	0.049	0.071	**0.062**
	F-measure ↑	0.723	0.782	0.856	0.900	0.917	0.916	**0.878**	**0.888**
PACSAL-S [27]	MAE ↓	0.194	0.211	0.142	0.116	0.087	0.093	0.152	**0.141**
	F-measure ↑	0.661	0.694	0.781	0.817	0.848	0.837	0.721	0.776
HKU-IS [26]	MAE ↓	0.139	0.145	0.079	0.064	0.046	0.042	**0.059**	**0.050**
	F-measure ↑	0.704	0.777	0.859	0.892	0.906	0.907	0.858	**0.872**
THUR-15K [28]	MAE ↓	0.148	0.147	0.066	0.097	0.077	0.087	0.096	0.094
	F-measure ↑	0.617	0.666	0.736	0.747	0.778	0.752	0.710	0.722

TABLE II: Comparison of unsupervised and weakly-supervised methods. SaliencyGAN is trained with 30% labelled images and 70% unlabelled images from MSRA10K. Results of the best performed method is shown in red. The SaliencyGAN method outperformed the weakly supervised method is shown in bold.

Dataset	Metric	Separate	Joint
DUT-TE [23]	MAE ↓	0.121	**0.095**
	F-measure ↑	0.633	**0.730**
ECSSD [25]	MAE ↓	0.108	**0.071**
	F-measure ↑	0.793	**0.878**
PACSAL-S [27]	MAE ↓	0.192	**0.152**
	F-measure ↑	0.657	**0.721**
HKU-IS [26]	MAE ↓	0.096	**0.059**
	F-measure ↑	0.772	**0.858**
THUR-15K [28]	MAE ↓	0.122	**0.096**
	F-measure ↑	0.638	**0.710**

TABLE III: Performance of SaliencyGAN networks that are separately trained against jointly trained using Algorithm 1. The better results obtained on each dataset are shown in bold.

VI. Conclusion

This paper introduces a semi-supervised SaliencyGAN model for SOD in modern vision based IoT systems. SaliencyGAN train a deep SOD CNN within its duo-GAN architecture and the associated learning procedure where the SOD CNN can be flexibly selected based on specific fog devices. We tested the proposed method in an simulated fog-based IoT environment. Experiments on multiple benchmark datasets have shown that the proposed model can achieve comparable performances with the fully trained supervised model using much less images. Simulated and ablation experiments also proved the ability of SaliencyGAN to precisely model image and saliency-map distribution using both labelled and unlabelled data. This enables fast adoption to a broad range of visual IoT applications and fog devices with various computing power.

References

[1] M. Aazam, K. A. Harras, and S. Zeadally, "Fog computing for 5g tactile industrial internet of things: Qoe-aware resource allocation model," *IEEE Transactions on Industrial Informatics*, 2019.

[2] H. Zhang, X. Cao, J. K. Ho, and T. W. Chow, "Object-level video advertising: an optimization framework," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 520–531, 2016.

[3] H. Liu, S. Chen, and N. Kubota, "Intelligent video systems and analytics: A survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1222–1233, 2013.

[4] X. Zheng and Z. Cai, "Real-time big data delivery in wireless networks: A case study on video delivery," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2048–2057, 2017.

[5] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 2083–2090.

[6] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 136–145.

[7] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[8] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems*, 2017, pp. 5767–5777.

[10] W. Zheng, H. Liu, B. Wang, and F. Sun, "Cross-modal surface material retrieval using discriminant adversarial learning," *IEEE Transactions on Industrial Informatics*, 2019.

[11] K. Muhammad, R. Hamza, J. Ahmad, J. Lloret, H. Wang, and S. W. Baik, "Secure surveillance framework for iot systems using probabilistic image encryption," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3679–3689, 2018.

[12] C. Piciarelli, D. Avola, D. Pannone, and G. L. Foresti, "A vision-based system for internal pipeline inspection," *IEEE Transactions on Industrial Informatics*, 2018.

[13] S. Dong, Z. Gao, S. Sun, X. Wang, M. Li, H. Zhang, G. Yang, H. Liu, and S. Li, "Holistic and deep feature pyramids for saliency detection," in *British Machine Vision Conference (BMVC)*, 2018.

[14] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 478–487.

[15] T. Wang, A. Borji, L. Zhang, P. Zhang, and H. Lu, "A stagewise refinement model for detecting salient objects in images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4019–4028.

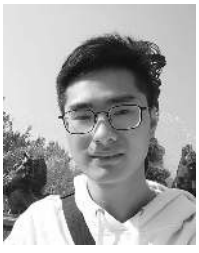
[16] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, "Fast interactive object annotation with curve-gcn," *arXiv preprint arXiv:1903.06874*, 2019.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

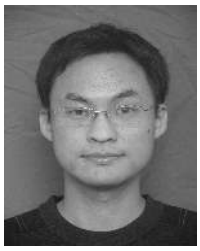
[18] H. Li, G. Li, and Y. Yu, "Rosa: Robust salient object detection against adversarial attacks," *IEEE transactions on cybernetics*, 2019.

[19] C. Esposito, X. Su, S. A. Aljawarneh, and C. Choi, "Securing collaborative deep learning in industrial applications within adversarial scenarios," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4972–4981, 2018.
[20] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
[21] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.
[22] C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2017, pp. 240–248.
[23] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3166–3173.
[24] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin, and X. Ruan, "Learning to detect salient objects with image-level supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 136–145.
[25] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1155–1162.
[26] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5455–5463.
[27] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 280–287.
[28] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu, "Salientshape: Group saliency in image collections," *The Visual Computer*, vol. 30, no. 4, pp. 443–453, 2014.
[29] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2334–2342.

Chengjia Wang (SM'09–M'15) received the M.Sc. degrees in vision and robotics from the European Erasmus Mundus Master Programme, in 2011, and the Ph.D. degree in machine learning-based medical image analysis from The University of Edinburgh, Edinburgh, UK. He joined the BHF Centre for Cardiovascular Science,The University of Edinburgh, as a Scientist in Machine Learning for Machine Learning, in 2016. His current research interests include computer vision, AI in medicine, and machine learning-based medical image analysis.



Shizhou Dong Shizhou Dong received the B.S. degree from Chongqing University of Posts and Telecommunications, China, in 2017. Currently he is a master student in Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. His work focuses on computer vision and deep learning architecture.



Xiaofeng Zhao is a full-time lecturer at the School of Management Engineering and Business, Hebei University of Engineering, China. He received his M.E in Computer Software and Theory from Hefei University of Technology. His research interests include artificial intelligence, information management, data mining and distributed computing.

Giorgos Papanastasiou is a post-doctoral Research Fellow at Edinburgh Imaging in Queen's Medical Research Institute of the University of Edinburgh. He received his PhD degree in Medical Physics and Mathematical modelling at the Clinical Research Imaging Centre and the British Heart Foundation-Centre for Cardiovascular Science of the University of Edinburgh, UK.

Heye Zhang (M'17) received the B.S. and M.E. degrees from Tsinghua University, Beijing, China, in 2001 and 2003, respectively, and the Ph.D. degree from the Hong Kong University of Science and Technology, Hong Kong, in 2007. Before Joining Sun Yat-sen University on 2018, he was a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Now He is leading a health informatics computing lab in School of Biomedical Engineering, Sun Yat-sen University. His research interests include cardiac electrophysiology and cardiac image analysis.

Guang Yang obtained his M.Sc. in Vision Imaging and Virtual Environments from the Department of Computer Science in 2006 and his Ph.D. on medical image analysis jointly from the CMIC, Department of Computer Science and Medical Physics in 2012 both from University College London. He is currently an honorary lecturer with the Neuroscience Research Centre, Cardiovascular and Cell Sciences Institute, St. George's, University of London. He is also an image processing physicist and honorary senior research fellow working at Cardiovascular Research Centre, Royal Brompton Hospital and also affiliate with National Heart and Lung Institute, Imperial College London. His research interests include: pattern recognition, machine learning, and medical image processing and analysis. His current research projects are funded by the British Heart Foundation.