Deep Co-saliency Detection via Stacked Autoencoder-enabled Fusion and Self-trained CNNs

Chung-Chi Tsai*, Kuang-Jui Hsu*, Yen-Yu Lin, *Member, IEEE,*
Xiaoning Qian, *Senior Member, IEEE* and Yung-Yu Chuang, *Member, IEEE*

***Abstract*—Image co-saliency detection via fusion-based or learning-based methods faces cross-cutting issues. Fusion-based methods often combine saliency proposals using a majority voting rule. Their performance hence highly depends on the quality and coherence of individual proposals. Learning-based methods typically require ground-truth annotations for training, which are not available for co-saliency detection. In this work, we present a two-stage approach to address these issues jointly. At the first stage, an unsupervised deep learning model with stacked autoencoder (SAE) is proposed to evaluate the quality of saliency proposals. It employs latent representations for image foregrounds, and auto-encodes foreground consistency and foreground-background distinctiveness in a discriminative way. The resultant model, SAE-enabled fusion (`SAEF`), can combine multiple saliency proposals to yield a more reliable saliency map. At the second stage, motivated by the fact that fusion often leads to over-smoothed saliency maps, we develop self-trained convolutional neural networks (`STCNN`) to alleviate this negative effect. `STCNN` takes the saliency maps produced by `SAEF` as inputs. It propagates information from regions of high confidence to those of low confidence. During propagation, feature representations are distilled, resulting in sharper and better co-saliency maps. Our approach is comprehensively evaluated on three benchmarks, including MSRC, iCoseg, and Cosal2015, and performs favorably against the state-of-the-arts. In addition, we demonstrate that our method can be applied to object co-segmentation and object co-localization, achieving the state-of-the-art performance in both applications.**

***Index Terms*—Co-saliency detection, stacked autoencoder, reconstruction residual, adaptive fusion, optimization, self-paced learning, CNNs.**

I. Introduction

CO-SALIENT object detection simulates human visual systems to search for visually attracting objects repetitively appearing across images. As an essential component of visual content understanding, it has become an inherent part in many applications, such as image co-segmentation [1]–[3], image co-localization [4] and content-aware compression [5]. Despite the significant progress on co-saliency detection [6]–[22], the general conclusion is still that no single model is sufficient for handling increasingly complex saliency detection of broad object categories.

To overcome this issue, saliency detection via proposal fusion has been a trend since it can combine the strengths of diverse saliency models while easing individual bias. Advanced fusion methods, e.g. [2], [6], [9], [20], [22], often adaptively rank the proposal quality before determining the weights for fusion. However, these methods judge the proposals' quality by measuring the degree of consistency with the other proposals. In other words, they assume the foreground regions from different saliency proposals have a high correlation; and thus they consider a proposal more reliable if its corresponding predictions agree with the group consensus. However, such an assumption may not hold if the adopted saliency proposals are not reliable or have substantial variations.

Another research trend is to employ deep convolutional models to automatically learn the discriminative features for salient object detection [17], [21], [23]–[29]. However, most off-the-shelf models require large-scale manual supervision for the ground truth and cannot address the task of co-saliency detection due to its unsupervised nature. We believe the concepts of fusion-based and deep-learning-based approaches can well complement each other if we can design a unified method such that their particular advantages can be transferred to help each other.

We confront this challenge by proposing a two-stage approach for robust co-saliency detection. At the first stage, we develop an unsupervised deep learning model, called stacked autoencoder-enabled fusion (`SAEF`), to evaluate and fuse multiple saliency proposals. The idea behind `SAEF` is simple: A saliency proposal for an image is considered good if its foreground can be well reconstructed by using object-like regions of other images while its background cannot. Specifically, SAEF learns a stacked autoencoder to reconstruct the object-like regions of an image, and apply the learned autoencoders across images to estimate not only foreground consistency but also foreground-background distinctiveness. In addition to image-level proposal evaluation, `SAEF` achieves better fusion by further exploring the complementary, co-saliency likelihood for region-level proposal evaluation. In brief, `SAEF` resolves the limitations of fusion-based and deep-learning-based methods. As an unsupervised model, it does not require supervisory data for training. It evaluates the quality of saliency proposals via discriminative reconstruction, and does not suffer from the difficulties caused by substantial variations

C.-C. Tsai is with the Department of Electrical and Computer Engineering, Texas A&M University, Texas 77843, USA and the Research Center for Information Technology Innovation, Academia Sinica, Taipei 115, Taiwan. E-mail: chungchi@tamu.edu

K.-J. Hsu and Y.-Y. Chuang are with the Research Center for Information Technology Innovation, Academia Sinica Taipei 115, Taiwan and the Department of Computer Science and Information Engineering, Nation Taiwan University, Taipei 106, Taiwan. E-mail: kjhsu@citi.sinica.edu.tw; cyy@csie.ntu.edu.tw

Y.-Y. Lin is with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan. E-mail: lin@cs.nctu.edu.tw

X. Qian is with the Department of Electrical and Computer Engineering, Texas A&M University, Texas 77843, USA. E-mail: xqian@ece.tamu.edu

* indicates equal contribution

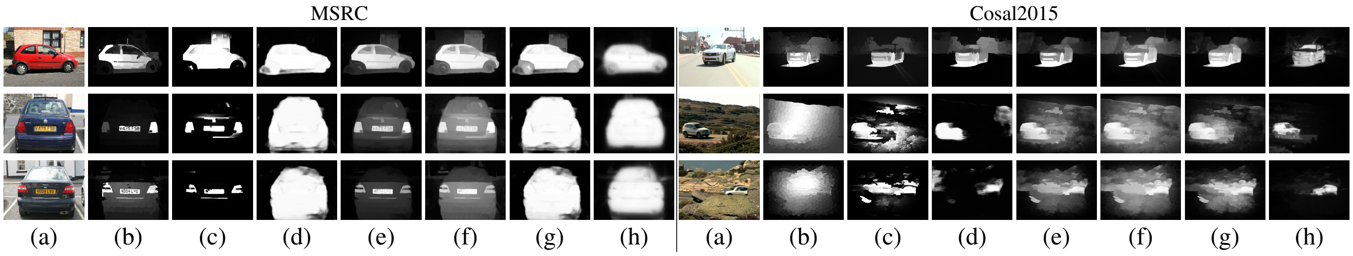


Fig. 1: Two examples of co-saliency detection. (a) Input images of the category car from the MSRC dataset (**left**) [30] and the Cosal2015 dataset (**right**) [15]. (b) ∼ (d) Three saliency proposals generated by GP [31], MST [32], and SVFSal [27] respectively. (e) Results generated by the map-wise proposal fusion method SACS [9]. (f) Results generated by the region-wise proposal fusion method CSSCF [2]. (g) Results generated by the proposed SAEF. (h) Refined results by the proposed STCNN.

or unreliable proposals.

Saliency maps generated by fusing multiple proposals are prone to be over-smoothed, and may inherit noise from proposals. At the second stage, we design self-trained convolutional neural networks (STCNN) to address the two issues. STCNN refines the saliency maps produced by SAEF. It propagates information from high-confidence regions to low-confidence ones in an iterative fashion while avoiding the refined saliency maps from being inconsistent with the original ones. The refined saliency maps look sharper with better preservation of object boundaries and with noise removed.

Fig. 1 shows two examples of co-saliency detection. The input images of the category car are displayed in Fig. 1(a). Three saliency proposals by using GP [31], MST [32], and SVFSal [33] are given in Fig. 1(b) ∼ (d), respectively. The proposals by SVFSal are of higher quality but are not consistent with the other two proposals. Fig. 1(e) and (f) show the co-saliency maps detected by map-wise [9] and region-wise [2] proposal fusion, respectively. The two fusion-based methods work based on majority consensus, and fail to assign a higher weight to the better proposal by SVFSal. Thus, their results in Fig. 1(e) and (f) are not satisfactory, and are even worse than the proposal by SVFSal. The proposed SAEF performs discriminative reconstruction for proposal evaluation. It derives a more plausible combination of the proposals, yielding much better results in Fig. 1(g). The proposed STCNN refines the saliency maps by using self-paced learning. As illustrated in Fig. 1(h), the refined saliency maps homogeneously highlight the whole objects and the background noise is greatly suppressed.

The main contribution of this work is two-fold. First, we propose stacked autoencoder-enabled fusion (SAEF) to tackle the limitations of fusion-based and learning-based co-saliency detection. SAEF carries out discriminative reconstruction for reliably measuring the quality of saliency proposals in an unsupervised manner. Thus, it saves human efforts to select higher quality saliency proposals to fuse and does not suffer from the problems caused by proposals with substantial variations. Second, the proposed STCNNs refine saliency maps by propagating information in a self-taught fashion, thereby learning the way to detect co-salient objects. The proposed method is evaluated on three representative and large-scale benchmarks, including the MSRC, iCoseg, and Cosal2015 datasets. Our method performs favorably against the state-of-the-arts in several tasks, including image co-saliency detection, as well as consequent object co-segmentation and co-localization.

The rest of this paper is organized as follows. Section II presents the literature review. We introduce the proposed SAEF and STCNN in Sections III-A and III-B, respectively. Our co-saliency detection method is evaluated in Section IV, and is applied to object co-segmentation and co-localization in Section V. Finally, we conclude the paper in Section VI.

II. Related Work

We review relevant topics to the development of our approach in this section, including saliency detection, co-saliency detection, and self-paced learning.

A. Saliency detection

Saliency detection aims to model human visual attention to identify distinct objects and segment them from an image. Conventional methods, e.g. [31], [32], [34]–[44], distinguish salient objects from backgrounds based on various low-level features. For instance, considering regions near image boundaries as background, several strategies, such as, low-rank matrix recovery theory [34], diffusion-based formulation [35], [36], minimum barrier distance [39], Markov random walks [37], or minimum spanning tree [32], etc., are utilized to measure the difference between the target superpixels and the background seeds for saliency prediction. To achieve better performance in cases that one or more of the boundaries happen to be adjacent to the foreground object, Li *et al.* [37] further integrate color contrast to remove the erroneous boundary which tends to have distinctive color distribution. Despite the efficiency, their unsupervised nature limits their performance once the images contain cluttered background and diverse object parts. To address this issue, supervised methods [23]–[28], [41] by using machine learning methodologies has been developed to accomplish salient object detection better. However, these methods rely on supervisory data annotations, which are costly and not available in general for saliency detection.

B. Co-saliency detection

Co-saliency detection is a weakly supervised extension to saliency detection. It leverages not only intra-image appearance evidence but also inter-image co-occurrence to locate common salient objects appearing in multiple images. Different strategies have been proposed for this task. *Bottom-up* methods utilize contrast hypothesis and different prior knowledge by either handcrafted features [6]–[11], [13], [20] or learned features [15], [16] to catch intra-image saliency as well as inter-image consistency. To further improve the performance, *fusion-based* methods merge several saliency models to exclude individual prediction bias while retaining the shared information. To this end, methods of this category fuse the saliency proposals generated by different models via fixed weight fusion [7], adaptive weight fusion [9], or region-wise adaptive fusion [2], [6], [20], [22]. Fusion-based methods typically work based on the assumption that plausible proposals are those sharing higher similarity with other proposals. Their performance drops when the assumption does not hold: the adopted saliency proposals have common prediction errors or large variations. *Deep-learning-based* methods [14], [17], [18], [29] are effective in distilling semantic object information in complex scenes, and have greatly enhanced co-saliency detection. However, these methods work in a supervised manner and require either a pre-trained deep model or labeled training data. Furthermore, the supervised setting also reduces their generalizability of handling objects of unseen categories.

Our SAEF tackles the cross-cutting issues of fusion-based and deep-learning-based methods. SAEF employs an unsupervised deep model to estimate the quality of each saliency proposal via auto-encoding both foreground consistency and foreground-background separation. It can more accurately identify the plausible proposals, and does not suffer from the unfavorable effects caused by fusion using majority voting. Besides, it does not rely on annotated training data and can detect salient objects of unseen categories.

C. Self-paced Learning

Kumar *et al.* [45] proposed self-paced learning (SPL) to imitate humans' learning behavior, namely starting to learn easier parts of a task and gradually considering more complex parts. Specifically, SPL associates each data sample with a weight. A self-spaced regularizer is attached to determine each weight value. Through sequential optimization, gradually increasing penalty on the regularizer includes more samples from easy to complex in training in a self-paced way. SPL has been widely used in various applications, such as matrix factorization [46], multimedia search [47], object tracking [48], image deblurring [49], action understanding [50], and co-saliency detection [16].

The method by Zhang *et al.* [16] is the most relevant to ours because it also adopts SPL for co-saliency detection. Different from ours, the SPL formulation in [16] is built on support vector machines (SVMs), and it treats feature extraction and co-saliency detection as separate steps. In contrast, our proposed SPL module STCNN is built on CNNs so that CNNs can jointly learn the relevant features and refine co-saliency detection in a self-paced fashion. The quality of the resultant saliency maps is hence greatly improved.

III. Proposed Method

This section describes our approach, which is composed of two components: stacked autoencoder-enabled fusion (SAEF) and self-trained convolutional neural networks (STCNNs). The former fuses saliency proposals and generates plausible saliency maps with unsupervised deep learning. The latter takes the saliency maps produced by SAEF as pseudo ground truth, and implements self-paced learning for saliency map refinement. Fig. 2 provides the flowchart of SAEF. The following two subsections detail SAEF and STCNN, respectively.

A. SAEF *for Proposal Fusion*

1) Problem Formulation: Given a set of N images $\mathcal{I} = \{I_n\}_{n=1}^N$ covering salient objects of the same category, we aim at detecting the salient objects in $\mathcal{I}$. As a fusion-based method, SAEF applies M existing saliency detection models, including [31]–[33], [37], [39], [40] in this work, to $\mathcal{I}$, and gets M saliency proposals $\{S_{n,m}\}_{m=1}^M$ for each image I_n. To abstract unnecessary details and extract the intrinsic structures at different scales, we hierarchically decompose each image I_n into K_n *segments* and T_n *superpixels*. Specifically, we derive initial coarse-level segments based on the algorithm in [51], and then group pixels into fine-level superpixels that can adhere to the boundary of the segments at the coarse level. In our experiments, we set the number of superpixels in each image to 200 and the number of pixels within each segment to be greater than 200. It follows that set $\mathcal{I}$ contains $K = \sum_n K_n$ segments and $T = \sum_n T_n$ superpixels in total. For proposal fusion, SAEF optimizes plausible weights $Y = [\mathbf{y}_1 \ \cdots \ \mathbf{y}_i \ \cdots \mathbf{y}_T] \in [0,1]^{M\times T}$, where vector $\mathbf{y}_i = [y_{i,1} \ y_{i,2} \ \ldots \ y_{i,M}]^\top \in [0,1]^M$ corresponds to superpixel i, to fuse the M saliency proposals in the domain of superpixels.

SAEF formulates the task of optimizing Y as an energy minimization problem over a graph $\mathcal{G} = (\mathcal{V} = \cup \, \mathcal{V}_n, \mathcal{E} = \cup \, \mathcal{E}_n)$, which encodes the spatial relationships among superpixels. Set $\mathcal{V}_n$ contains T_n nodes, one for each superpixel in image I_n. Edge e_{ij} is added to $\mathcal{E}_n$ for linking nodes v_i and v_j if superpixels i and j are spatially connected in image I_n. Edge e_{ij} is associated with a weight $a_{ij} = \exp(-\|\mathbf{v}_i - \mathbf{v}_j\|^2)$, where $\mathbf{v}_i$ and $\mathbf{v}_j$ are the deep features of superpixels i and j, respectively. How $\mathbf{v}_i$ and $\mathbf{v}_j$ are extracted will be given later. *Graph Laplacian* $L \in \mathbb{R}^{T\times T}$ of $\mathcal{G}$ is then computed based on the affinity matrix $A = [a_{ij}] \in \mathbb{R}^{T\times T}$.

Before designing the objective function for optimizing Y, we investigate the potential foreground areas of each image I_n. To this end, B object proposals $\{f_{n,b}\}_{b=1}^B$ are generated by applying the scheme in [52] to I_n, where B is set to 350 here. To further explore the object mask corresponding to each proposal $f_{n,b}$, we consider superpixel v_i belongs to $f_{n,b}$ if 1) it is fully covered by $f_{n,b}$ or 2) it is partially covered by $f_{n,b}$ and the area ratio $|v_i \cap f_{n,b}|/|f_{n,b}|$ is larger than $|f_{n,b}|/|I_n|$. The corresponding mask of $f_{n,b}$ is defined to be composed of all the superpixels belonging to $f_{n,b}$. The feature representation of the mask is denoted by $\mathbf{f}_{n,b}$ and is

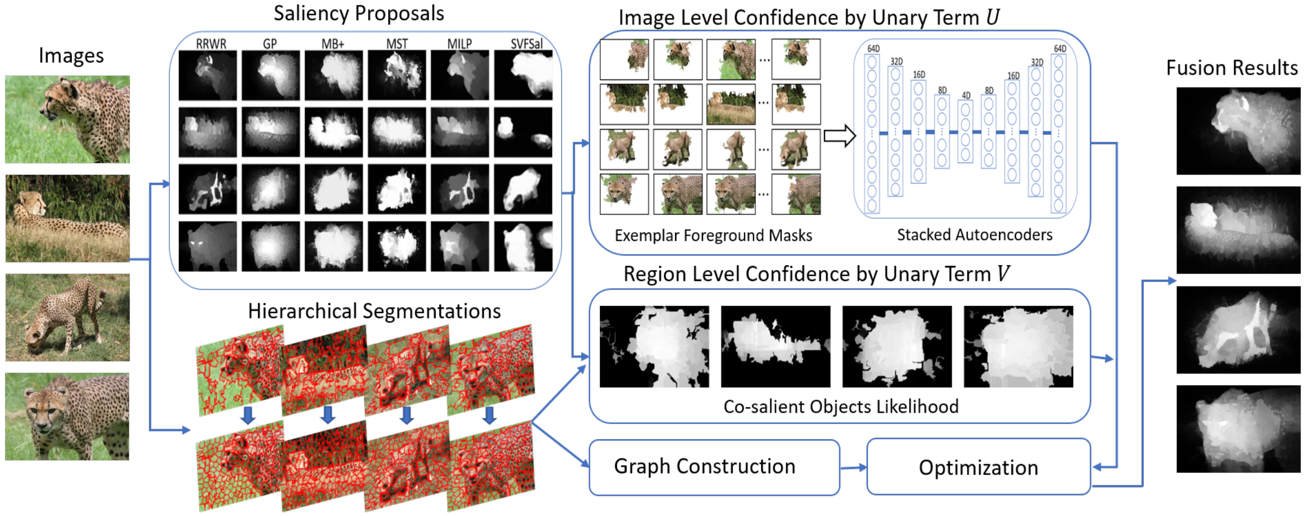

Images
Saliency Proposals
RRWR
GP
MB+
MST
MILP
SVFSal
Image Level Confidence by Unary Term U
64D
32D
16D
8D
4D
8D
16D
32D
64D
Exemplar Foreground Masks
Stacked Autoencoders
Fusion Results
Region Level Confidence by Unary Term V
Hierarchical Segmentations
Co-salient Objects Likelihood
Graph Construction
Optimization

Fig. 2: Overview of SAEF. SAEF collects multiple saliency proposals, extracts superpixels, and constructs a graph structure for the input images. It formulates co-saliency detection as an optimization problem with an objective function considering both image- and region-level confidence. After completing the optimization, saliency maps are produced.

yielded by max-pooling the feature vectors of all superpixels it covers. The procedure is repeated for each object proposal. The collected foreground masks of image I_n are $\mathcal{F}_n = \{\mathbf{f}_{n,b}\}_{b=1}^{B}$, which represent our initial estimation of the salient object in I_n. Figure 2 provides a schematic illustration summarizing the flowchart of SAEF. By transferring useful information from object proposals containing the object segments as well as the estimated co-salient object likelihood, our proposed fusion method possesses more robustness in finding the optimal fusion weights; thus consistent improvement can be achieved when applying the proposed optimization model to existing saliency approaches.

2) Objective Function: SAEF seeks the optimal weights $Y = [\mathbf{y}_1 \ \cdots \ \mathbf{y}_T] \in \mathbb{R}^{M \times T}$ for superpixel-wise saliency proposal fusion by minimizing the following objective function defined over $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$$\begin{aligned} \min_{Y} & \sum_{v_i \in \mathcal{V}} (U(\mathbf{y}_i) + \lambda_1 V(\mathbf{y}_i)) \\ & + \lambda_2 \sum_{e_{ij} \in \mathcal{E}} B(\mathbf{y}_i, \mathbf{y}_j) + \lambda_3 \, \|Y\|_2^2 \qquad (1) \\ \text{s.t.} \quad & \|\mathbf{y}_i\|_1 = 1, \mathbf{y}_i \geq \mathbf{0}, \ \text{for } 1 \leq i \leq T, \end{aligned}$$

where $\mathbf{0}$ is an all-zero vector, λ_1, λ_2 and λ_3 are three positive constants. The unary term $U(\mathbf{y}_i)$ is the primary element in SAEF. It refers to the proposals' reconstruction errors from SAE (stacked autoencoder) and image-wisely determines the quality of each proposal. The auxiliary unary term $V(\mathbf{y}_i)$ takes the *co-salient object likelihood* into account and can superpixel-wisely refine the weights for fusion. Pairwise term $B(\mathbf{y}_i, \mathbf{y}_j)$ encourages the spatial smoothness of the derived weights. Lastly, $||Y||_2^2$ is a regularization term. The terms $U(\mathbf{y}_i)$, $V(\mathbf{y}_i)$, and $B(\mathbf{y}_i, \mathbf{y}_j)$ are detailed as follows.

a) On Designing Unary Term $U(\mathbf{y}_i)$: This term evaluates the quality of each saliency proposal for the image covering superpixel i based on a stacked autoencoder (SAE) [53] representation, encoding both foreground consistency and foreground-background distinctiveness, to determine a plausible weight vector $\mathbf{y}_i$ for fusion.

Recall that we collect B potential object masks for each image I_n, extract and denote their features by $\mathcal{F}_n = \{\mathbf{f}_{n,b}\}_{b=1}^{B}$. For I_n, we learn an SAE H_{θ_n} by minimizing the cross-entropy between the inputs in $\mathcal{F}_n$ and the reconstructed outputs, where θ_n is the learned parameter set. In this way, this SAE can reconstruct the estimated foreground masks of I_n. The procedure is repeated for every image. A total of N SAEs $\{H_{\theta_n}\}_{n=1}^{N}$ are obtained.

Considering image I_n and its mth proposal $S_{n,m}$, I_n can be partitioned into the foreground and the background sub-images by Otsu's thresholding, denoted as $I_{n,m}^f$ and $I_{n,m}^b$. We use the same way to represent the sub-image $I_{n,m}^f$. Namely, the feature representation $\mathbf{x}_{n,m}^f$ of $I_{n,m}^f$ is yielded by max-pooling the feature vectors of the superpixels belonging to $I_{n,m}^f$. The feature representation $\mathbf{x}_{n,m}^b$ of $I_{n,m}^b$ is obtained similarly.

Assume that the mth proposal for image I_n is of high quality. The reconstruction error by applying SAE $H_{\theta_{n'}}$ to the detected foreground of I_n, i.e. $\|\mathbf{x}_{n,m}^f - H_{\theta_{n'}}(\mathbf{x}_{n,m}^f)\|$, is expected to be low since images I_n and $I_{n'}$ have common foreground objects. In addition, the reconstruction error $\|\mathbf{x}_{n,m}^b - H_{\theta_{n'}}(\mathbf{x}_{n,m}^b)\|$ is probably high when we feed SAE $H_{\theta_{n'}}$ with the estimated background of I_n. To jointly consider inter-image foreground similarity and foreground-background distinctiveness, we compute the ratio between the foreground and background reconstruction errors

$$\hat{g}_{n,m} = \frac{\sum_{n'=1}^{N} \|\mathbf{x}_{n,m}^b - H_{\theta_{n'}}(\mathbf{x}_{n,m}^b)\|^2}{\sum_{n'=1}^{N} \|\mathbf{x}_{n,m}^f - H_{\theta_{n'}}(\mathbf{x}_{n,m}^f)\|^2}. \qquad (2)$$

To take other proposals into account, the image-level score

$g_{n,m}$ of the mth proposal for image I_n is calculated by

$$g_{n,m} = \frac{\exp(\hat{g}_{n,m})}{\sum_{m'=1}^{M} \exp(\hat{g}_{n,m'})}. \tag{3}$$

A penalty variable $r_{i,m} = (1 - g_{n,m})$ is introduced if superpixel i belongs to image I_n. The first term in Eq. (1) is defined by

$$\sum_{v_i \in \mathcal{V}} U(\mathbf{y}_i) = \sum_{i=1}^{T} \mathbf{r}_i^\top \mathbf{y}_i = tr(R^\top Y), \tag{4}$$

where $\mathbf{r}_i = [r_{i,1} \cdots r_{i,M}]^\top$ and $R = [\mathbf{r}_1 \cdots \mathbf{r}_T]$.

b) On Designing Unary Term $V(\mathbf{y}_i)$: This term refines the fusion weights $\mathbf{y}_i$ on superpixel i locally. It is designed based on the formula of co-salient object likelihood

$$\textit{Co-saliency} = \textit{Saliency} \times \textit{Correspondence}. \tag{5}$$

Suppose superpixel i belongs to image I_n. For the *saliency* part, we transfer the objectness score $\psi_{n,b}$ suggested by [52] from every object mask $\mathbf{f}_{n,b}$ covering superpixel i to superpixel i, i.e.

$$O(v_i) = \sum_{b=1}^{B} \psi_{n,b} \delta(v_i \in \mathbf{f}_{n,b}), \tag{6}$$

where δ is the indicator function.

We also explore the location information by using the functional properties of coarse-level segments. Unlike fine-level superpixels that have grid alike structure, coarse-level segments adhere better to the image content variation; and are usually long boundary connected in the background area while having smaller fragmented regions on the object area. Since segments near the image center, Ctr_n, more likely belong to the foreground while those overlapping with the set of the image boundary pixels, Bou_n, tend to be covered by background. Suppose that superpixel i is covered by the kth segment u_k. The location prior of superpixel i is defined as

$$\begin{aligned} L(v_i) = &\mathcal{N}(\|\texttt{cord}(u_k) - Ctr_n\|^2 \mid 0, \sigma^2) \\ &\times \exp\big(\frac{-2|u_k \cap Bou_n|}{\texttt{per}(u_k)}\big), \end{aligned} \tag{7}$$

where $\texttt{cord}(u_k)$ and $\texttt{per}(u_k)$ are the center and the perimeter of segment u_k, respectively. $\mathcal{N}$ is the normal distribution with σ set to the geometric mean of the image width and height.

For $O(v_i)$ in Eq. (6) and $L(v_i)$ in Eq. (7), we linearly scale each of them to $[0, 1]$ by taking all superpixels in the same image into account. Then, the *saliency* score of superpixel i is yielded by averaging the corresponding scaled values.

For the *correspondence* part, we examine if there are strong correspondences of superpixel i in other images. To this end, we apply Otsu's thresholding method to divide the superpixels of each image I_n into foreground and background according to their *saliency* parts estimated above. Recall that all superpixels are represented by the deep features. A *Gaussian mixture model* (GMM) $\boldsymbol{\theta}_f$ with five components is fit to the foreground superpixels of all images. Meanwhile, a five-component GMM $\boldsymbol{\theta}_{b,n}$ is fit to the background superpixels of I_n, for $n = 1, 2, ..., N$. The *correspondence* score of superpixel i is defined as

$$C(v_i) = \frac{p(\mathbf{v}_i|\boldsymbol{\theta}_f)}{p(\mathbf{v}_i|\boldsymbol{\theta}_f) + \sum_{n=1}^{N} p(\mathbf{v}_i|\boldsymbol{\theta}_{b,n}) \delta(v_i \in I_n)}, \tag{8}$$

where $\mathbf{v}_i$ is the deep features of superpixel i, $p(\mathbf{v}_i|\boldsymbol{\theta}_f)$ and $p(\mathbf{v}_i|\boldsymbol{\theta}_{b,n})$ are the probabilities estimated by GMMs $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_{b,n}$, respectively.

$C(v_i)$ is also linearly scaled to $[0, 1]$ and then multiplied by the *saliency* score to compute Eq. (5) as the co-saliency prior $CS(v_i)$ of superpixel i. Let $s_{i,m}$ be the mean saliency value of the mth saliency proposal on superpixel i. We prefer a saliency proposal consistent with the co-saliency prior. Let ϕ_n be the Otsu's threshold over the co-saliency prior of all superpixels in I_n. The score of saliency proposal m on superpixel i is defined as

$$l_{i,m} = \frac{\exp(-\|\delta(CS(v_i) \geq \phi_n) - s_{i,m}\|^2)}{\sum_{m'=1}^{M} \exp(-\|\delta(CS(v_i) \geq \phi_n) - s_{i,m'}\|^2)}. \tag{9}$$

With $\mathbf{l}_i = [l_{i,1} \cdots l_{i,M}]^\top$, the second term in Eq. (1) is set to

$$\sum_{v_i \in \mathcal{V}} V(\mathbf{y}_i) = \sum_{i=1}^{T} (1 - \mathbf{l}_i)^\top \mathbf{y}_i. \tag{10}$$

c) On Designing Pairwise Term $B(\mathbf{y}_i, \mathbf{y}_j)$: This pairwise term is added to encourage the spatial smoothness of Y on $\mathcal{G}$:

$$\sum_{e_{ij} \in \mathcal{E}} B(\mathbf{y}_i, \mathbf{y}_j) = \sum_{e_{ij} \in \mathcal{E}} a_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 = tr(YLY^\top), \tag{11}$$

where a_{ij} is the weight of e_{ij} and L is the graph Laplacian of $\mathcal{G}$.

3) Optimization and Implementation Details: With the terms $U(\mathbf{y}_i)$ in Eq. (10), $V(\mathbf{y}_i)$ in Eq. (4), and $B(\mathbf{y}_i, \mathbf{y}_j)$ in Eq. (11), the constrained optimization problem in Eq. (1) can be solved by *quadratic programming* (QP). We optimize it with the CVX solver [54], and get the weights for fusion Y^*. The saliency maps $\{\hat{S}_n\}_{n=1}^N$ for images $\{I_n\}_{n=1}^N$ are then produced.

In our implementation of the stacked autoencoders (SAE), we adopt the 5-layer network architecture used in [53], but reduce the numbers of neurons in the five layers to 64, 32, 16, 8, and 4 respectively due to the limited training data. For each image, its features are generated by applying ResNet50 [55] to it. Specifically, we up-sample and concatenate the feature maps in layers conv1_relu, res2c_relu, res3d_relu, res4f_relu, and res5c_relu to yield the 3904-d *hypercolumn* representation. The feature vector of each superpixel is calculated by max-pooling over the region it covers.

B. `STCNN` *for Saliency Map Refinement*

We introduce self-trained CNNs (`STCNN`) for saliency map refinement in this section.

1) Problem Formulation: The saliency maps produced by `SAEF` are prone to being over-smoothed and may contain inherited noise from proposals. `STCNN` addresses these issues by introducing self-paced learning. It propagates information from high-confidence regions to low-confidence ones, and progressively refines the saliency maps.

`STCNN` is a CNN-based model with two network streams, f_g and f_l. Both take an image as input and predict its saliency map. While f_g approximates the saliency maps that `SAEF` produces as the pseudo ground truth, f_l carries out self-paced learning for iterative saliency map refinement. The objective for training `STCNN` is

$$\ell(\mathbf{w}_g, \mathbf{w}_l; \mathcal{I}) = \ell_g(\mathbf{w}_g; \mathcal{I}) + \ell_l(\mathbf{w}_l; \mathcal{I}), \tag{12}$$

where loss functions ℓ_g and ℓ_l guide the training of f_g and f_l respectively, and will be detailed later. Sets $\mathbf{w}_g$ and $\mathbf{w}_l$ cover the learnable parameters of f_g and f_l, respectively. After optimizing Eq. (12), the refined saliency map S_n of image I_n is produced via $S_n = f_g(I_n; \mathbf{w}_g) \times f_l(I_n; \mathbf{w}_l) = S_n^g \times S_n^l$.

a) On Designing Loss ℓ_g: This term aims to detect the common salient objects by approximating the saliency maps $\{\hat{S}_n\}$, treated as the pseudo ground truth, generated by `SAEF`, and ℓ_g is defined as

$$\ell_g(\mathbf{w}_g; \mathcal{I}) = \sum_{n=1}^{N} \sum_{p \in I_n} Q_n(p) |S_n^g(p) - \hat{S}_n(p)|^2, \tag{13}$$

where p is the index of the pixels in I_n and $S_n^g = f_g(I_n; \mathbf{w}_g)$ is the saliency map generated by f_g. $S_n^g(p)$ and $\hat{S}_n(p)$ are the saliency values of S_n^g and $\hat{S}_n$ at pixel p, respectively. $Q_n(p)$ indicates the importance of pixel p. We partition the pixels in $\hat{S}_n$ into two categories, salient and non-salient, by using the mean value of $\hat{S}_n$ as the threshold. $Q_n(p)$ is introduced to address the potential size unbalance between the two categories. Let ρ be the ratio between salient pixels and all pixels. $Q_n(p)$ is set to $1 - \rho$ if pixel p is categorized as salient, and ρ otherwise. In this way, the pixels in the two categories contribute equally in Eq. (13).

The loss function in Eq. (13) is optimized by considering all images $\{I_n\}_{n=1}^N$ simultaneously. Compared with `SAEF`, f_g can better learn the visual properties shared among salient objects while excluding the individual backgrounds, to achieve better performance.

b) On Designing Loss ℓ_l: The term ℓ_l in Eq. (12) leverages self-paced learning (SPL) to iteratively identify and learn from high-confidence regions, and propagate the information to better predict low-confidence regions in saliency maps. It is defined as

$$\begin{aligned} \ell_l(\mathbf{w}_l, &\{\mathbf{M}_n, \mathbf{V}_n\}_{n=1}^N; \mathcal{I}) = \\ &\sum_{n=1}^{N} \sum_{p \in I_n} \mathbf{V}_n(p) |S_n^l(p) - \mathbf{M}_n(p)|^2 - \gamma \mathbf{V}_n(p), \quad (14) \\ s.t.\ & \mathbf{V}_n(p) \in [0, 1], \mathbf{M}_n(p) \in \{0, 1\}, \forall n, p, \end{aligned}$$

where $S_n^l = f_l(I_n; \mathbf{w}_l)$ and the constant γ controls the learning pace. For image I_n, the auxiliary variable $\mathbf{M}_n$ denotes the estimated co-saliency mask. Each pixel p is associated with a latent weight variable $\mathbf{V}_n(p)$ weighting the corresponding loss. The first term in Eq. (14) measures the consistency between the predicted saliency maps and the estimated masks while the second term favors selecting easy over complex samples (pixels here). Namely, a sample with less loss is considered *easy*, so it is learned with a higher priority and vice versa. In sum, minimizing ℓ_l in Eq. (14) decreases the weighted training loss together with the negative ℓ_1-norm regularizer.

Eq. (14) consists of three sets of optimization variables, $\mathbf{w}_l$, $\{\mathbf{M}_n\}_{n=1}^N$, and $\{\mathbf{V}_n\}_{n=1}^N$. Because directly optimizing Eq. (14) is difficult, we instead adopt an alternating iterative strategy to optimize $\mathbf{w}_l$, $\{\mathbf{M}_n\}_{n=1}^N$, and $\{\mathbf{V}_n\}_{n=1}^N$. At each iteration, one of the three variables is optimized while keeping the others fixed in an alternating fashion. The iterative procedure is repeated until convergence.

On optimizing $\mathbf{w}_l$: We fix $\{\mathbf{M}_n, \mathbf{V}_n\}_{n=1}^N$. The optimization problem in Eq. (14) is reduced to

$$\begin{aligned} &\ell_l(\mathbf{w}_l, \{\mathbf{M}_n, \mathbf{V}_n\}_{n=1}^N; \mathcal{I}) = \\ &\qquad \sum_{n=1}^{N} \sum_{p \in I_n} \mathbf{V}_n(p) |S_n^l(p) - \mathbf{M}_n(p)|^2. \quad (15) \end{aligned}$$

Stochastic gradient descent (SGD) is adopted to optimize the parameters $\mathbf{w}_l$ of CNNs f_l.

On optimizing $\mathbf{w}_l$: By fixing $\mathbf{w}_l$ and $\{\mathbf{V}_n\}_{n=1}^N$, the optimization problem in Eq. (14) becomes

$$\begin{aligned} &\ell_l(\mathbf{w}_l, \{\mathbf{M}_n, \mathbf{V}_n\}_{n=1}^N; \mathcal{I}) = \\ &\qquad \sum_{n=1}^{N} \sum_{p \in I_n} \mathbf{V}_n(p) |S_n^l(p) - \mathbf{M}_n(p)|^2, \quad (16) \\ &s.t.\ \mathbf{M}_n(p) \in \{0, 1\}, \forall n, p. \end{aligned}$$

It is obvious that the optimal $\mathbf{M}_n(p)$ takes value 0 if $S_n^l(p) \leq 0.5$, and 1 otherwise.

On optimizing $\{\mathbf{V}_n\}_{n=1}^N$: When fixing $\mathbf{w}_l$ and $\{\mathbf{M}_n\}_{n=1}^N$, as shown in [45], the global optimum $\{\mathbf{V}_n\}_{n=1}^N$ can be obtained via

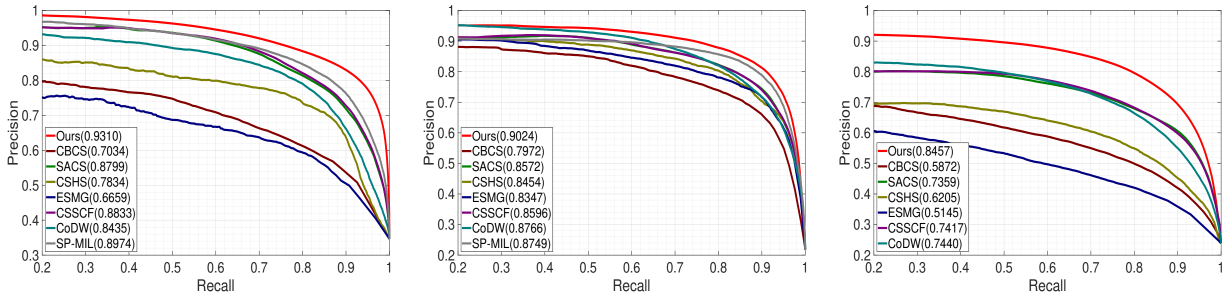
$$\mathbf{V}_n(p) = \begin{cases} 1, & \text{if } |S_n^l(p) - \mathbf{M}_n(p)|^2 < \gamma, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

2) Optimization and Implementation Details: Consider the optimization of `STCNN` in Eq. (12). We first optimize Eq. (13) with backward propagation to obtain optimum $\mathbf{w}_g^*$ for f_g, and then optimum $\mathbf{w}_l^*$ for f_l is obtained by optimizing Eq. (14) iteratively. Prior to running alternating optimization, we initialize $\{\mathbf{M}_n, \mathbf{V}_n\}$ with saliency maps $\{\hat{S}_n\}$ `SAEF` produces as follows

$$\mathbf{V}_n(p) = \begin{cases} 1, & \text{if } \hat{S}_n(p) > \mu_n + \sigma_n, \\ 1, & \text{if } \hat{S}_n(p) < \mu_n - \frac{\sigma_n}{4}, \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

$$\mathbf{M}_n(p) = \begin{cases} 1, & \text{if } \hat{S}_n(p) > \mu_n + \sigma_n, \\ 0, & \text{if } \hat{S}_n(p) < \mu_n - \frac{\sigma_n}{4}, \\ \times, & \text{otherwise,} \end{cases} \tag{19}$$

where × denotes *don't-care*. μ_n and σ_n are the mean and standard deviation of the saliency values in $\hat{S}_n$.



MSRC iCoseg Cosal2015

Fig. 3: The performance of various methods in PR curves on different datasets. The numbers in parentheses denote AP.

Algorithm 1 Optimization Procedure

```
Input: A collection of images, $\{I_n\}_{n=1}^{N}$; Saliency proposals, $\{S_{j,m}\}_{m,n=1}^{m=M,n=n}$, Max epochs, $T$;
    /* SAEF begins */
    Generate fusion weights $\{Y^*_{n,m}\}$ via optimizing Eq. (1);
    Compile saliency maps $\{\hat{S}_n\}$ via weights $\{Y^*_{n,m}\}$;
    /* SAEF ends */
    /* STCNN begins */
    Learn stream $\mathbf{w}^*_g$ via optimizing Eq. (13);
    Initialize $\{\mathbf{V}_n\}$ and $\{\mathbf{M}_n\}$;
    for $i \leftarrow 1, 2, \ldots, T$ do
        Update stream $\mathbf{w}^*_l$ by solving Eq. (15);
        Update $\{S^l_n = f_l(I_n; \mathbf{w}^*_l)\}$;
        Update $\{\mathbf{M}_n\}$ via binarizing $\{S^l_n\}$ with a threshold 0.5;
        Update $\{\mathbf{V}_n\}$ via Eq. (17);
    end for
    Generate $\{S^g_n = f_g(I_n; \mathbf{w}^*_g)\}$ with the learned stream $\mathbf{w}^*_g$;
    Generate $\{S^l_n = f_l(I_n; \mathbf{w}^*_l)\}$ with the learned stream $\mathbf{w}^*_l$;
    Produce saliency maps $\{S_n = S^g_n \times S^l_n\}$ for images $\{I_n\}$;
    /* STCNN ends */
    Post-precess $\{S_n\}$ via DenseCRFs;
Output: Co-saliency maps, $\{S_n\}$
```

Pixel p with $V_n(p) = 1$ represents that it can be confidently assigned to either the salient regions ($\mathbf{M}_n(p) = 1$) or the background ($\mathbf{M}_n(p) = 0$). It is taken into account at the current epoch. Others with $V_n(p) = 0$ are ambiguous, so they are currently ignored. ADAM [56] is chosen as the optimization solver for its rapid convergence. In practice, for each image I_n at each epoch, the mask $\mathbf{M}_n$ and the latent varialbes $\mathbf{V}_n$ are updated only when $\mathbf{w}_l$ is stable enough, namely the squared error between the predicted saliency map and the estimated mask less than 0.1^2 in our cases. The maximum number of epochs is set to 60. The gradient derivation with respect to the optimization variables is straightforward, so it is omitted here. With $\mathbf{w}^*_g$ and $\mathbf{w}^*_l$, the refined saliency map S_n by STCNN is then calculated by $S_n = S^g_n \times S^l_n = f_g(I_n; \mathbf{w}^*_g) \times f_l(I_n; \mathbf{w}^*_l)$.

Please note that the trivial solution: $S^l(p) = 0, M(p) = 0, V(p) = 1$, is indeed the global optimum to the loss in Eq. (14). But the adopted alternating optimization can avoid the trivial solution: $S^l_n(p) = 0, \mathbf{M}_n(p) = 0$ and $\mathbf{V}_n(p) = 1, \forall n, p$. We first solve it by learning STCNN via minimizing the joint objective in Eq. (12), which is a combination of the two loss functions in Eq. (13) and Eq. (14). In this way, the trivial solution is no longer the optimal one. Nevertheless, we found that given a proper initialization (described in Eqs. (18) and (19)) before minimizing Eq. (14), each of three sub-optimization problems in Eq. (15) $\sim$ Eq. (17) searches for the optimum solution with other variables fixed, and does not fall into the trivial solution. For easier optimization, we do not optimize the joint objective in Eq. (12), but optimize Eq. (13) and Eq. (14) sequentially. Finally, we implement STCNN using MatConvNet [57]. The same network architecture, i.e. VGG-16 [58] setting of FCN [59], is adopted for both network streams, f_g and f_l. We replace the activation function *softmax* in the last layer with the *sigmoid* function. The learning rate is fixed as 10^{-5}. The weight decay, momentum, and batch size are set to 0.0005, 0.9, 5, respectively.

3) Post-processing using DenseCRFs: Spatial coherence and object boundary preservation of the saliency maps generated by STCNN can be further enhanced. Following the previous work [25], [26], DenseCRFs [60] is adopted to post-process each saliency map. In our cases, the unary and the pairwise terms in DenseCRFs are set to S_n and bilateral filtering, respectively. Please refer to Li and Yu's paper [25], [26] for the definitions of the two terms in detail. After post-processing, the inferred posterior probabilities of being salient yield the final saliency map. In this work, the public DenseCRFs code implemented by Li and Yu [25] is used. To conclude our method, the whole optimization process including SAEF and STCNN is summarized in Algorithm 1.

IV. Experimental Results

In this section, we first describe the datasets and evaluation metrics. Then, we compare our method with state-of-the-art methods, and investigate contributions of individual components by conducting ablation studies.

A. Datasets

We evaluated the proposed approach on three public benchmark datasets: *iCoseg* [61], *MSRC* [30], and *Cosal2015* [15]. *iCoseg* consists of 38 groups of total 643 images. The images of *iCoseg* contain single or multiple similar objects in various poses and sizes with complex backgrounds. *MSRC* contains 7 groups of total 240 images. Compared to *iCoseg*, co-salient

Method	Year	Setting	MSRC			iCoseg			Cosal2015		
			AP	F_β	S_α	AP	F_β	S_α	AP	F_β	S_α
LEGS [23]	CVPR2015	SI+S	0.8479	0.7701	0.6997	0.7924	0.7473	0.7529	0.7339	0.6926	0.7068
DHS [24]	CVPR2016	SI+S	0.8907	0.8186	0.7815	0.8791	0.8216	0.8428	0.7940	0.7366	0.7843
DCL [25]	CVPR2016	SI+S	0.9065	0.8259	0.7742	0.9003	0.8444	0.8606	0.7814	0.7388	0.7596
DSS [26]	CVPR2017	SI+S	0.8700	0.8313	0.7435	0.8802	0.8386	0.8483	0.7745	0.7510	0.7582
UCF [27]	ICCV2017	SI+S	0.9217	0.8114	0.8175	0.9292	0.8261	0.8754	0.8080	0.7197	0.7797
Amulet [28]	ICCV2017	SI+S	0.9219	0.8159	0.8162	0.9395	0.8381	0.8937	0.8201	0.7387	0.7863
MSC-NET [29]	MM2017	SI+S	0.9035	0.8419	0.7673	0.8845	0.8378	0.8518	0.8328	0.7683	0.7994
DIM [14]	TNNLS2016	CS+S	-	-	-	0.8773	0.7918	0.7583	-	-	-
UMLBF [18]	TCSVT2017	CS+S	0.9160	0.8410	-	-	-	-	0.8210	0.7120	-
RRWR [37]	CVPR2015	SI+US	0.8127	0.7534	0.6653	0.7986	0.7784	0.7022	0.6647	0.6636	0.6628
GP [31]	ICCV2015	SI+US	0.8200	0.7422	0.6844	0.7821	0.7495	0.7198	0.6851	0.6580	0.6721
MB+ [39]	ICCV2015	SI+US	0.8367	0.7817	0.7200	0.7868	0.7706	0.7272	0.6715	0.6693	0.6732
MST [32]	CVPR2016	SI+US	0.8057	0.7491	0.6460	0.8019	0.7659	0.7292	0.7099	0.6672	0.6681
MILP [40]	TIP2017	SI+US	0.8334	0.7776	0.6871	0.8182	0.7883	0.7514	0.6802	0.6737	0.6757
SVFSal [33]	ICCV2017	SI+US	0.8669	0.7934	0.7688	0.8376	0.8056	0.8271	0.7467	0.7123	0.7607
CBCS [8]	TIP2013	CS+US	0.7034	0.5910	0.4801	0.7972	0.7408	0.6580	0.5872	0.5583	0.5444
SACS [9]	TIP2014	CS+US	0.8799	0.8027	0.7341	0.8572	0.8048	0.7783	0.7359	0.7089	0.7170
CSHS [11]	SPL2014	CS+US	0.7834	0.7118	0.6661	0.8454	0.7549	0.7502	0.6205	0.6186	0.5918
ESMG [13]	SPL2015	CS+US	0.6659	0.6245	0.5804	0.8347	0.7766	0.7677	0.5145	0.5120	0.5454
CSSCF [2]	TMM2016	CS+US	0.8833	0.8136	0.7626	0.8596	0.7929	0.7686	0.7417	0.6997	0.6950
CoDW [15]	IJCV2016	CS+US	0.8435	0.7724	0.7129	0.8766	0.7985	0.7500	0.7440	0.7048	0.6482
SP-MIL [16]	TPAMI2017	CS+US	0.8974	0.8029	0.7687	0.8749	0.8143	0.7715	-	-	-
MVSRC [19]	TIP2017	CS+US	0.8530	0.7840	-	0.8680	0.8100	-	-	-	-
SAEF	/	CS+US	0.8850	0.8110	0.7758	0.8561	0.7967	0.7808	0.7401	0.7052	0.7269
Ours	/	CS+US	0.9310	0.8397	0.8062	0.9024	0.8452	0.8216	0.8457	0.7814	0.7703

TABLE I: Quantitative comparison with 20 methods on three benchmark datasets. “SI” and “CS” denote the single-image saliency and co-saliency methods, respectively. “S” and “US” indicate the supervised and unsupervised methods, respectively. Among the “US” methods, the top three results are marked in red, green and blue, in the order. Our fusion method SAEF mostly outperforms the other two fusion methods SACS and CSSCF. With self-training CNNs, our final result leads all the competing unsupervised methods in most cases and has comparable performance with the supervised approaches.

objects in *MSRC* exhibit less pose or viewing angle variation; however, it contains different colors and shapes. Thus, the *MSRC* appears to be almost equally difficult as the *iCoseg* dataset. Lastly, *Cosal2015* is a more recent and the most challenging dataset among three so far. It has 50 groups and a total of 2015 images containing significant poses and sizes, appearance variations and even more complex backgrounds.

B. Evaluation Metrics

To evaluate the performance of co-saliency detection, we adopt two commonly used metrics: *average precision* (AP) and *F-measure* (F_β), as well as a newly proposed metric: *structure measure* (S_α) [62]. AP is the area under the Precision-Recall (PR) curve by comparing the ground truth with the binary masks produced by varying the saliency map threshold continuously in the range of $[0, 1]$. Meanwhile, with a self-adaptive threshold $T = \mu + \sigma$, where μ and σ denote the mean and standard deviation of the saliency map respectively, F_β*-measure* is computed by the harmonic mean of the precision and recall values: $F_\beta = \frac{(1+\beta^2)\times precision\times recall}{\beta^2\times precision+recall}$, with the imposed weight $\beta^2 = 0.3$ to emphasize more on recall as suggested in [15], [16], [35], [63]. In addition to the aforementioned pixel-based metrics, a region-based image quality measure, *structure measure* (S_α) [62], is adopted to evaluate the spatial structure similarity of saliency maps based on both region-aware structural similarity S_r and object-aware structural similarity S_o, defined as $S_\alpha = \alpha * S_r + (1-\alpha) * S_o$, where $\alpha = 0.5$ following [62]. Specifically, to evaluate the region-aware structural similarity measure, the full saliency map is first divided into K non-overlapping blocks. Then the region similarity of each block $ssim(k)$ is computed by comparing with the ground truth based on the product of three components: luminance comparison, contrast comparison, and structure comparison. For each component,the similarity measure is defined similarly as Pearson correlation [62]. With $ssim(k)$, a different weight w_k is assigned to each block based on the foreground region each block covers, and it is formulated as: $S_r = \sum_{k=1}^{K} w_k \times ssim(k)$. Meanwhile, the object-aware structural similarity is designed with respect to two characteristics: *sharp foreground-background contrast* and *uniform saliency distribution* by measuring the mean pixel values of the final saliency map in foreground ($\bar{x}_{FG}$) & background ($\bar{x}_{BG}$) regions and the corresponding standard deviation values of foreground ($\sigma_{x_{FG}}$) & background ($\sigma_{x_{BG}}$) regions (defined by the ground truth), respectively. Specifically, $S_o = (O_{FG}+O_{BG})/2$, $O_{FG} = \frac{2\bar{x}_{FG}}{(\bar{x}_{FG})^2+1+2\lambda\times\sigma_{x_{FG}}}$, and O_{BG} is similarly computed. This metric is proposed to alleviate the flaw of widely used pixel-based measures, for example, AP, AUC, F_β, or even the recently introduced generalized F-measure F_β^w [64], as it is observed that any foreground map

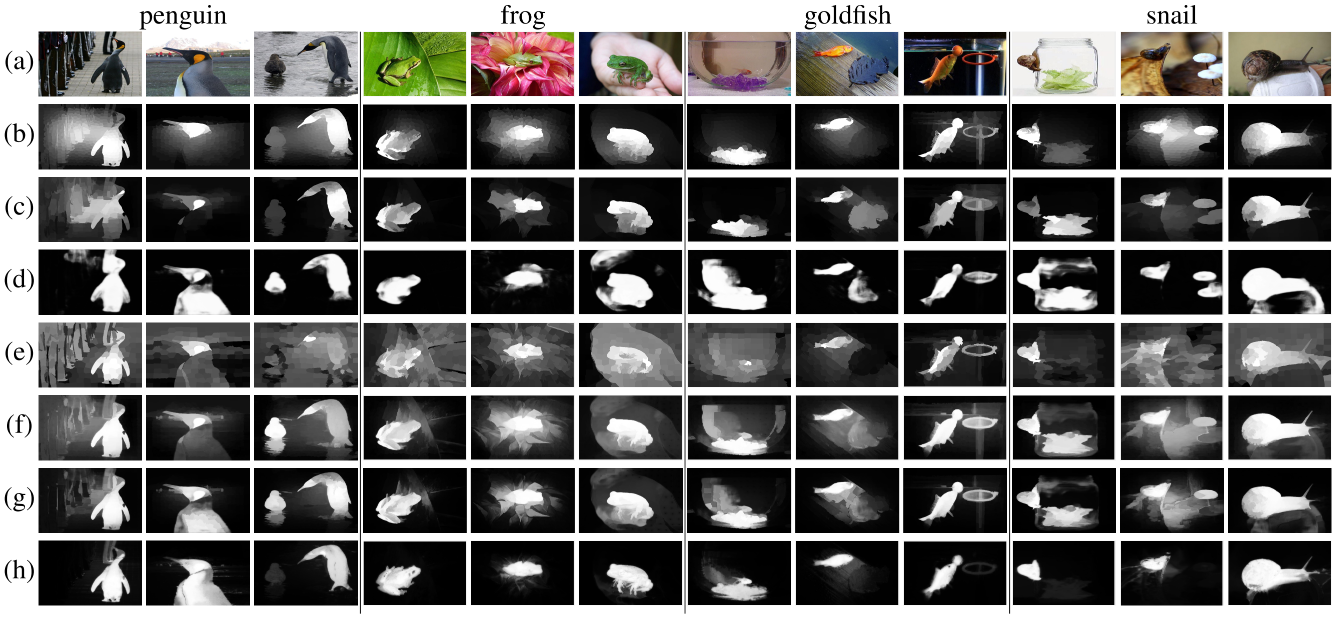
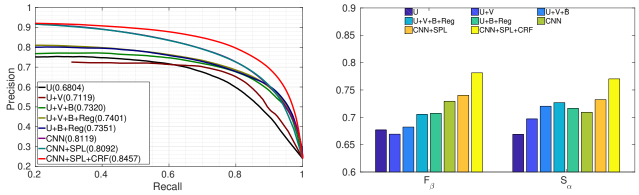


Fig. 4: Visual comparison with state-of-the-art methods. (a) Images from four image groups of the Cosal2015 dataset for co-saliency detection. (b)∼(h) Saliency maps generated by different approaches, including (b) `GP` [31], (c) `MILP` [40], (d) `SVFSal` [33], (e) `CoDW` [15], (f) `CSSCF` [2], (g) `SAEF`, and (h) `Ours`.



(a) PR-curves and AP

(b) F_β and S_α

Fig. 5: Ablation studies on the Cosal2015 dataset in terms of (a) the PR curves and AP (in parentheses), (b) F_β and S_α.

that more preserves the entire object structure can help the machine to learn a more complete object information.

C. Comparison with the State-of-the-Arts

To have a thorough comparison with state-of-the-art methods, we divide them into four groups, i.e. the unsupervised saliency [31]–[33], [37], [39], [40] and co-saliency [2], [8], [9], [11], [13], [15], [16], [19] detection methods as well as supervised saliency [23]–[29] and co-saliency [14], [18] detection methods. The overall performance statistics are summarized in TABLE I and Fig. 3. Please note that all compared supervised single-image saliency detection methods are CNN-based. We reproduced the experimental results using the publicly available source code with default parameters provided by the authors. For methods without releasing source code, we either evaluated on their released results (`SP-MIL` [16], `CoDW` [15] and `DIM` [14]), or directly reported the numbers in their papers (`UMLBF` [18] and `MVSRC` [19]). Note that the results of `CBCS` [8], `SACS` [9], `CSHS` [11] and `ESMG` [13] may not be exactly the same as those reported in [19]. Regarding the resolutions of input images, we modify the released code of `CBCS` [8], `SACS` [9], `CSHS` [11] and `ESMG` [13] for a consistent comparison. In our setting, the image resolutions in MSRC, iCoseg, and Cosal2015 are resized to 320 × 320, 512 × 512, and 512 × 512, respectively. For a fair comparison, we use the same resolutions for all competing methods, and resize the results to the original resolutions for evaluation. When evaluating `SACS` [9], we use the same saliency proposals as ours, instead of those originally used in `SACS` [9].

The precision-recall (PR) curves by our method and seven competing co-saliency detection methods on three different datasets are shown in Figure 3. The overall quantitative result is reported in TABLE I. With the same unsupervised setting, our method leads both the single-image saliency detection and co-saliency detection methods by a large margin. Moreover, by leveraging unsupervised deep learning and self-paced learning, our method even surpasses many supervised CNN-based single-image saliency methods that exploit object annotations. Last but not least, compared with the supervised co-saliency method `DIM` [14] that employs stack denoising autoencoder (SDAE) and `UMLBF` [18] that similarly applies adaptive feature learning for co-saliency detection, our method outperforms them without requiring expensive object annotations.

To gain insights into the quantitative results, in addition to the results in Figure 1, Figure 4 shows example saliency maps on four groups from the most challenging co-saliency detection dataset: Cosal2015. Single-image saliency detection methods generally cannot give satisfactory results. For instance, methods `GP` and `MILP` relying on specific hand-crafted cues inevitably produce many false positives in the first image of the Penguin class and miss the majority of penguin's body in the second image. Furthermore, without jointly exploiting the common objects in multiple images,

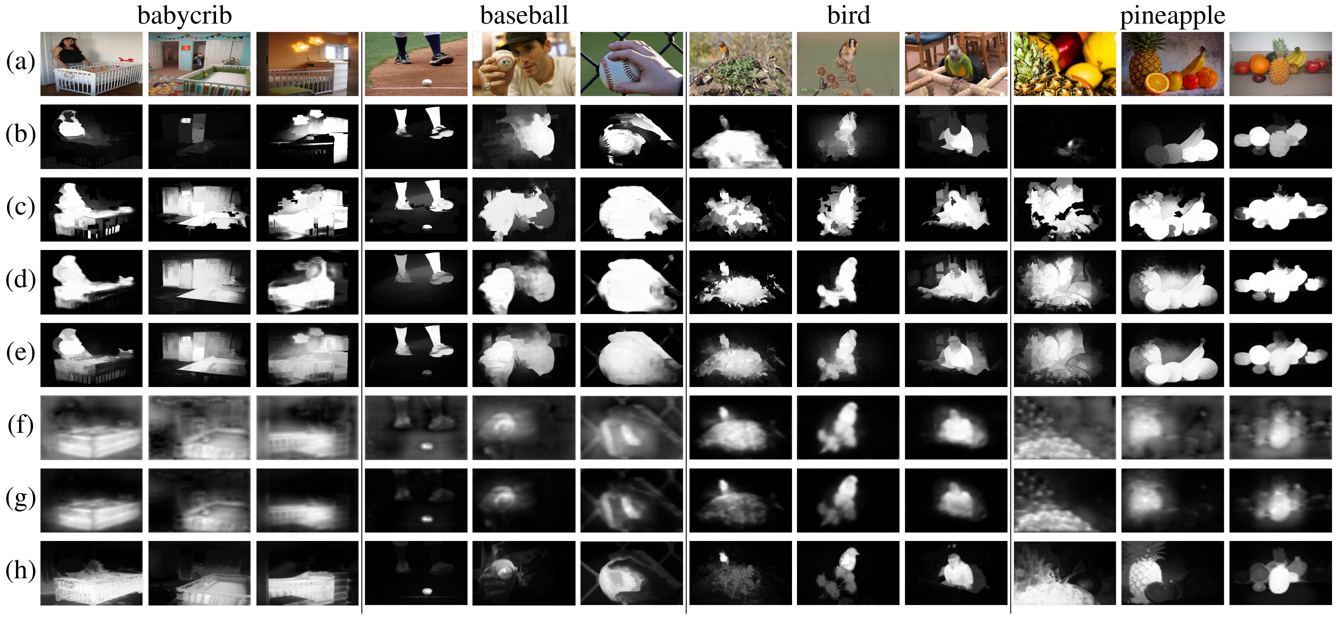


Fig. 6: Visual illustration of ablation studies. (a) Images from four groups for co-saliency detection. (b)∼(h) Co-saliency maps generated by different combinations of components, including (b) U, (c) $U+V$, (d) $U+V+B$, (e) $U+V+B+Reg$ (SAEF), (f) SAEF+CNN, (g) SAEF+CNN+SPL, (h) SAEF+CNN+SPL+DenseCRFs (Ours), respectively.

Method	MSRC			iCoseg			Cosal2015		
	AP	F_β	S_α	AP	F_β	S_α	AP	F_β	S_α
CSSCF [2] w/o DCRF	0.8833	0.8136	0.7626	0.8596	0.7929	0.7686	0.7417	0.6997	0.6950
CoDW [15] w/o DCRF	0.8435	0.7724	0.7129	0.8766	0.7985	0.7500	0.7440	0.7048	0.6482
SAEF w/o DCRF	0.8850	0.8110	0.7758	0.8561	0.7967	0.7808	0.7401	0.7052	0.7269
Ours w/o DCRF	0.9272	0.8493	0.7896	0.8695	0.7994	0.7808	0.8092	0.7402	0.7324
CSSCF [2] w/ DCRF	0.8990	0.8104	0.7811	0.8751	0.8186	0.7990	0.7500	0.7048	0.7170
CoDW [15] w/ DCRF	0.8734	0.7819	0.7560	0.9005	0.8260	0.7906	0.7710	0.7224	0.6871
SAEF w/ DCRF	0.8986	0.8137	0.7917	0.8693	0.8200	0.7953	0.7545	0.7171	0.7373
Ours w/ DCRF	0.9310	0.8397	0.8062	0.9024	0.8452	0.8216	0.8457	0.7814	0.7703

TABLE II: Performance comparison of ours and two best competing methods before and after using DenseCRFs (DCRF) for post-processing on three datasets. The top two results are marked in red and green, respectively.

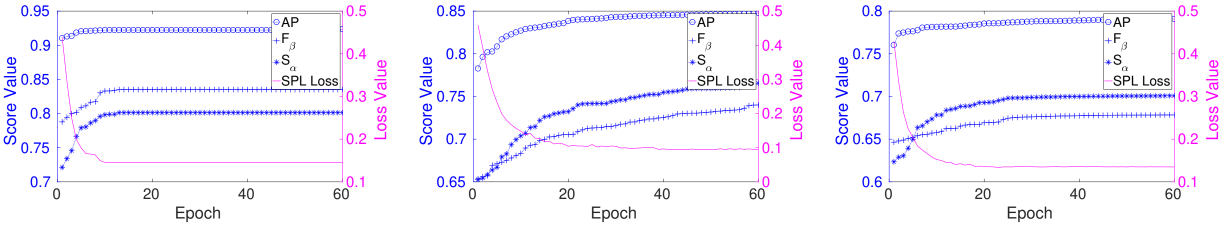
Method	Worst Proposal Duplication			Noisy Proposals		
	AP	F_β	S_α	AP	F_β	S_α
SACS [9]	0.8659 (-0.0140)	0.7759 (-0.0268)	0.7030 (-0.0311)	0.8485 (-0.0314)	0.7875 (-0.0202)	0.7007 (0.0334)
CSSCF [2]	0.8666 (-0.0167)	0.7899 (-0.0237)	0.7381 (-0.0245)	0.8561 (-0.0272)	0.7978 (-0.0158)	0.6832 (0.0794)
SAEF	0.8800 (-0.0050)	0.8001 (-0.0109)	0.7611 (-0.0147)	0.8689 (-0.0161)	0.8074 (-0.0036)	0.7729 (0.0029)

TABLE III: Performance and the drop in parentheses of three fusion-based methods under the unfavorable effects of *worst proposal duplication* and *noisy proposals*. The top two results are marked in red and green, respectively.

single-image saliency detection methods cannot exclude the visually salient objects that do not repetitively appear in other images. For instance, although the CNN-based single-image saliency detection method SVFSal can better delineate object boundaries, it often includes unrelated regions. As an example, the bird on the left-hand side of the third penguin image is wrongly taken as part of the co-salient object. Next, results of CoDW show that significant intra- and inter-object variations can sometimes mislead co-saliency detection and lead to results even worse than the single-image saliency detection methods. Though more relevant images bring more prosperous and shared information to explore in co-saliency detection, the problem is also more challenging as it needs to cope with potential variations across images. The fusion-based approach CSSCF deals with large inter-object variations by fusing the saliency proposals from the methods GP, MILP and SVFSal. It boosts the performance and surpasses the method CoDW. However, as mentioned above, it relies on the group consensus and can not discriminatively put more weight to the best saliency proposal. Our proposed SAEF generates better results than CSSCF by overcoming the inherent group biasing issue. Finally, our two-stage approach elegantly integrates a self-trained CNN guided by SAEF and gives sharper and more homogeneous saliency detection results by successfully filtering out the background noise and recovering the omissions.

Method	MSRC			iCoseg			Cosal2015		
	AP	F_β	S_α	AP	F_β	S_α	AP	F_β	S_α
`RRWR` [37]+STCNN	0.8804	0.8164	0.7660	0.8264	0.7508	0.7518	0.7689	0.6968	0.6879
`GP` [31]+STCNN	0.8642	0.8113	0.7474	0.7884	0.7119	0.7303	0.7502	0.6734	0.6614
`MB+` [39]+STCNN	0.8693	0.8138	0.7581	0.8052	0.7347	0.7522	0.7630	0.6889	0.6795
`MST` [32]+STCNN	0.8735	0.8161	0.7285	0.8172	0.7405	0.7475	0.7746	0.7018	0.6868
`MILP` [40]+STCNN	0.8864	0.8316	0.7685	0.8251	0.7394	0.7536	0.7659	0.6934	0.6916
`SVFSal` [33]+STCNN	0.9057	0.8345	0.7897	0.8437	0.7628	0.7980	0.8012	0.7295	0.7471
`SAEF`+STCNN	0.9272	0.8493	0.7896	0.8695	0.7994	0.7808	0.8092	0.7402	0.7324

TABLE IV: Quantitative comparison on three benchmark datasets by applying `STCNN` to the saliency proposals produced by six existing methods and `SAEF`. The top three results are marked in red, green, and blue, respectively. Note that these results are not post-processed with DenseCRFs.



(a) MSRC (b) iCoseg (c) Cosal2015

Fig. 7: Learning curves of our method during optimizing Eq. (14) on three datasets. Three blue dotted lines with the plus, circle, and asterisk signs represent the values of AP, F_β, and S_α at each epoch, respectively. The magenta solid line shows the loss function value of Eq. (14) at each epoch.

D. Ablation Studies

1) Energy term contribution: Fig. 5 reports ablation studies with different metrics to investigate contributions from individual energy terms of `SAEF` and from each component in the proposed `STCNN` network i.e. CNN (f_g), self-paced learning (SPL, f_l), and DenseCRFs (D). From AP and F_β scores, it is clear that the results improve progressively by adding individual energy terms, U, V, and B, into the objective function in Eq. (1). In addition, we compare the results of `SAEF` with and without hierarchical segmentation. The hierarchical segmentation is only adopted in the unary term V. For evaluating `SAEF` without hierarchical segmentation, we set λ_1 in Eq. (1) to 0. From the results of $U + V + B + Reg$ (with hierarchical segmentation) to $U + B + Reg$ (without hierarchical segmentation), the performance measured in *structure measure* and *average precision* drops because the hierarchical segmentation helps preserve the object structure. `STCNN` further boosts the detection results by combining CNN's object recognition capability with SPL dealing with the limited quantity of pseudo ground truth under the unsupervised learning setting. By integrating the merit of DenseCRFs, our method achieves superior results. The progressive improvement is not as evident for the metric S_α that measures the local structure similarity of the detected objects to the ground truth. The major reason is that some background regions badly predicted by SAEF will lead to the fuzzier maps generated by CNNs, which makes S_α lower.

Fig. 6 shows the co-saliency maps that visually illustrate the ablation study. Initially, by using only the image-wise confidence computed from the stacked autoencoder, the results (Fig. 6(b)) tend to bias toward a saliency map that indicates only apparent objects. By adding the region-wise confidence computed from the co-salient object likelihood, many missed regions are recovered (Fig. 6(c)). Furthermore, by adding the pairwise term that promotes smoothness, the resultant saliency maps are smoothed out (Fig. 6(d)). Lastly, after adding the regularization term, we obtain the best fusion result (Fig. 6(e)). However, as mentioned above, the drawback of fusion is that the outcome is limited by the adopted saliency proposals. Fortunately, after further integration with CNNs by propagating information from regions with high confidence, as shown in Fig. 6(f), the fusion results are gradually improved by emphasizing the common salient regions, but still left some blur background regions in some images. Fig. 6(g) shows that self-paced learning improves the results by reducing irrelevant backgrounds. Finally, DenseCRFs help yield sharper and more complete co-saliency maps as shown in Fig. 6(h).

2) Effectiveness of SAEF: The proposed SAEF suffers less from the difficulties caused by substantial image variations or unreliable proposals as indicated in TABLE III. The proposed `SAEF` and two fusion-based methods `SACS` [9] and `CSSCF` [2] fuse six proposals in the paper. We evaluate and compare the three methods in the experiment where the effects of *unreliable proposals* and *substantial variations* are exacerbated by duplicating the relatively lower-quality proposal `MST` 4 times and adding Gaussian noise with zero mean and variance 0.1 to the 5 lower-quality proposals, respectively. The results, as well as the performance drops of the three methods on the MSRC dataset, are given in TABLE III. It is clear that the proposed `SAEF` is more robust to the two unfavorable effects and has significantly smaller performance drops than `SACS` [9] and `CSSCF` [2] (about 1/2 $\sim$ 1/20 in most cases).

3) Performance without DenseCRFs: Using DenseCRFs for post-processing improves the cosaliency detection per-

Stage	Operation		Time (second)
SAEF	Unary term U construction	Object proposal generation	4.4790
		Proposal feature extraction (GPU)	1.0709
		Autoencoder optimization (GPU)	9.6839
		Image-level score generation via Eq. (2) and Eq. (3)	1.2605
	Unary term V construction	Coarse-level segments generation	0.0875
		Coarse-level segment feature extraction (GPU)	1.9843
		Objectness score generation via Eq. (6) and Eq. (7)	1.6287
		Correspondence score generation via Eq. (8)	10.5294
		Region-level score generation via Eq. (9)	0.3404
	Pairwise term B construction	Fine-level superpixel generation	3.8045
		Fine-level superpixel feature extraction (GPU)	2.0848
		Graph Laplacian L generation via Eq. (11)	1.0986
	Optimization of Eq. (1) with CVX		7.4604
STCNN	Optimization of Eq. (13)		18.6368
	Optimization of Eq. (14)		18.8270
Post-processing	DenseCRFs		0.3847

TABLE V: Average execution time of each component of our method on an image.

Method	CBCS	SACS	CSHS	ESMG	CSSCF	Ours
Time (s)	4.25	2.31	33.08	2.47	5.53	81.10

TABLE VI: Average execution time on an image.

formance of our method. We show the ablation studies and comparison with two state-of-the-art methods [2], [15] in TABLE II. It can be observed that our methods, SAEF and Ours (SAEF+STCNN), without using DenseCRFs still outperforms [2], [15] on the three datasets. By applying DenseCRFs to all methods, our methods also have the superior results.

4) Saliency proposals with STCNN*:* The proposed STCNN can also be applied to saliency proposals yielded by not only SAEF but also other methods, and further enhance the results. To demonstrate the effectiveness of SAEF, we compare the results generated by applying STCNN to the saliency proposals produced by SAEF and existing methods. TABLE IV reports the comparison results. It can be observed that compared with these proposals by the existing methods, SAEF can produce high-quality co-saliency maps, which serve as the input to STCNN to achieve better performance.

5) Convergence analysis: In Fig. 7, we plot the objective function values of Eq. (14) and the corresponding performance indices of our method at each epoch on the three datasets. Although the alternating iterative strategy is adopted to optimize the variables in Eq. (14), we can observe that the proposed method can converge rapidly and the performance is gradually improved along the optimization process. Since the performance and the objective function values do not change significantly after the 60th epoch, we set the maximal number of epochs to 60 in our experiments. In this way, the execution time to optimize Eq. (14) is about 18.8270 seconds per image.

6) Running time analysis: In TABLE V, we list the average execution time of each component of our method on an image. The execution time is measured on a workstation with one 3.7GHz 8-core CPU, 64GB memory, and a GTX Titan X GPU. The code is implemented in a mix of CUDA, MATLAB, and C without any code optimization. From TABLE V, we can observe that the most time-consuming part is the optimization of STCNN. However, from the optimization curves shown in Fig. 7, the computation cost can be further reduced via using less epochs, e.g. 20, because the performance doesn't significantly improve after the 20th epoch. In addition, we compare the execution time of our proposed method and other methods with released code, and show the comparison results in TABLE VI. Although our method is not as efficient as the competing methods, it can achieve much better performance as reported in TABLE I.

V. Applications

In this section, we apply the proposed approach to two applications, object co-segmentation and object co-localization. As suggested in [4], we convert our co-saliency maps to the results of object co-segmentation and object co-localization via the GrabCut algorithm and a thresholding method, respectively. In the following, we present two applications on the iCoseg and Cosal2015 datasets.

A. Object co-segmentation

Because values in the co-saliency maps are real-valued, following the previous work for object co-segmentation [2], [65], the GrabCut algorithm is used to generate the binary co-segmentation masks, where the unary terms are initialized with our estimated co-saliency maps. Given the estimated co-saliency maps, we use the GrabCut toolbox implemented in [65] to produce the results.

1) Evaluation metrics: We adopt two standard measures, *precision* ($\mathcal{P}$) and *Jaccard index* ($\mathcal{J}$), to evaluate the performance of object co-segmentation. Precision measures the percentage of correctly segmented pixels including both object and background pixels. Jaccard index is the ratio of the intersection area of the detected objects and the ground truth to their union area. The background pixels are taken into account in precision, so the images with larger background areas tend to have a better performance in precision. Therefore, precision may not faithfully reflect the quality of object co-segmentation results. Compared with precision, Jaccard index is considered more reliable to measure the quality of results. It provides more appropriate evaluation as it only focuses on objects.

2) Results: In TABLE VII, we compare our co-segmentation results with those generated by the state-of-the-art co-segmentation methods including CSC [65], MFC [66], GMS [67], GSP [68], MRW [69], SGCCCS [70], CSCS [71],

Method	Year	iCoseg		Cosal2015	
		$\mathcal{P}$	$\mathcal{J}$	$\mathcal{P}$	$\mathcal{J}$
CSC [65]	ICCV2013	88.3	0.66	82.3	0.37
MFC [66]	CVIU2015	72.0	0.40	67.2	0.33
GMS [67]	ICIP2014	89.2	0.64	83.5	0.54
GSP [68]	ICIP2015	89.5	0.65	83.6	0.54
MRW [69]	CVPR2015	91.2	0.70	72.8	0.45
SGCCCS [70]	CVPR2015	90.8	0.70	-	-
CSCS [71]	CVPR2017	85.9	0.58	83.1	0.53
QGFCE [4]	TMM2018	91.8	0.72	-	-
CBCS [8]	TIP2013	86.7	0.57	81.7	0.42
SACS [9]	TIP2014	90.8	0.70	87.0	0.60
CSHS [11]	SPL2014	89.4	0.65	79.4	0.51
ESMG [13]	SPL2015	88.0	0.65	76.6	0.41
CSSCF [2]	TMM2016	91.9	0.72	85.1	0.56
DIM [14]	TNNLS2016	90.6	0.69	-	-
CoDW [15]	IJCV2016	89.5	0.68	83.6	0.58
SP-MIL [16]	TPAMI2017	87.9	0.67	-	-
Ours	/	93.1	0.75	90.5	0.68

TABLE VII: Performance of object co-segmentation on two datasets. The numbers in red, green, and blue indicate the best, the second best and the third best results, respectively.

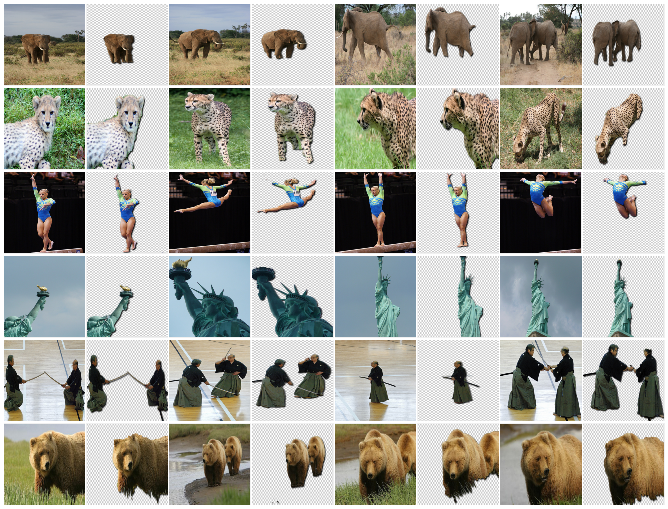


Fig. 8: Co-segmentation results generated by our approach on the iCoseg dataset. In the six examples (rows), the common object categories are elephant, cheetah, gymnastics, Statue of Liberty, kendo, and brown bear, respectively.

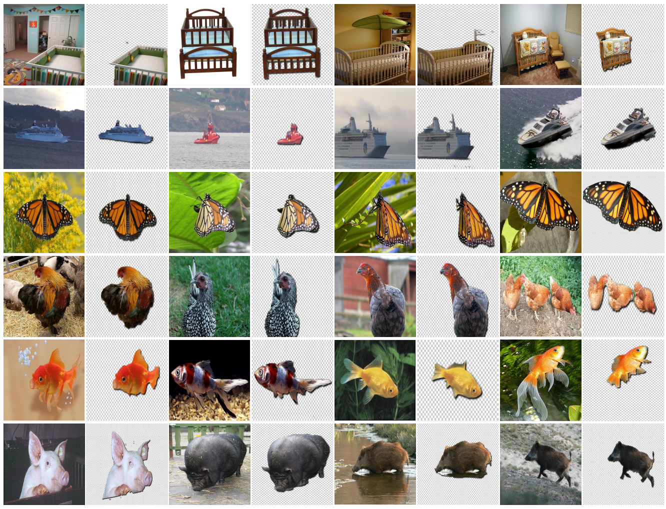


Fig. 9: Co-segmentation results generated by our approach on the Cosal2015 dataset. In the six examples (rows), the common object categories are babycrib, boat, butterfly, chook, goldenfish, and pig, respectively.

and QGFCE [4] on the iCoseg and Cosal2015 datasets. In addition to the methods for object co-segmentation, we also compare the state-of-the-art co-saliency results that are binarized via GraphCut, including CBCS [8], SACS [9], CSHS [11], ESMG [13], CSSCF [2], DIM [14], CoDW [15], and SP-MIL [16].

TABLE VII shows that the proposed approach achieves the state-of-the-art performance compared to the competing methods. This table also shows that good co-saliency maps can benefit co-segmentation. Compared to the co-segmentation methods, such as MFC [66], MRW [69], SGCCCS [70], CSCS [71], our method does not use any complex optimization process but only the simple GrabCut algorithm, and can achieve better performance. Compared with the co-saliency methods, our method achieves better performance owing to the better estimated co-saliency maps. Fig. 8 and Fig. 9 show some co-segmentation results on the iCoseg and Cosal2015 datasets, respectively. Our method can generate promising object segmentation results under different types of intra-class variations, such as colors, shapes, poses, and background clutters on both datasets.

B. Object co-localization

Following the method in [4], we first binarize the saliency maps into the binary masks using a threshold $T = \mu + 0.3 \times \sigma$, where μ and σ denote the mean and standard deviation of the saliency map, respectively. Then, for each image, we extract a single bounding box by fitting that box around the largest connected component in the binary mask.

1) Evaluation metrics: Following the previous image co-localization work [4], [72]–[74], the metric, correct localization (CorLoc), is taken for evaluating object co-localization. CorLoc is defined as the percentage of images correctly localized according to the PASCAL criterion: $\frac{B_p \bigcap B_{gt}}{B_p \bigcup B_{gt}} > 0.5$, where B_p and B_{gt} are the predicted box and the ground-truth box, respectively.

2) Results: We compare the proposed method with several existing methods on the iCoseg and Cosal2015 datasets, and report their performance in TABLE VIII. The competing methods include the state-of-the-art object co-localization algorithms: CLRW [72], UODL [73], DDT [74], DFF [75], and QGFCE [4]. In addition to the above methods, we also compare state-of-the-art co-saliency-based results whose bounding boxes are generated with the same scheme as what we adopt. These co-saliency methods include CBCS [8], SACS [9], CSHS [11], ESMG [13], CSSCF [2], DIM [14], CoDW [15], and SP-MIL [16].

From TABLE VIII, our method achieves the best performance among these competing methods on the iCoseg and Cosal2015 datasets. The results confirm that high-quality co-saliency maps are also helpful for object co-localization.

Method	Year	iCoseg	Cosal2015
CLRW [72]	CVPR2014	46.9	48.3
UODL [73]	CVPR2015	37.0	32.4
DDT [74]	IJCAI2017	30.4	31.6
DFF [75]	ECCV2018	40.8	52.3
QGFCE [4]	TMM2018	68.5	55.5
CBCS [8]	TIP2013	57.9	44.6
SACS [9]	TIP2014	76.1	64.1
CSHS [11]	SPL2014	66.6	55.0
ESMG [13]	SPL2015	67.1	46.2
DIM [14]	TNNLS2016	71.2	-
CoDW [15]	IJCV2016	72.7	63.1
SP-MIL [16]	TPAMI2017	70.6	-
Ours	/	82.8	74.0

TABLE VIII: Performance of object co-localization on the two datasets. The numbers in red, green, and blue indicate the best, the second best, and the third best results, respectively.

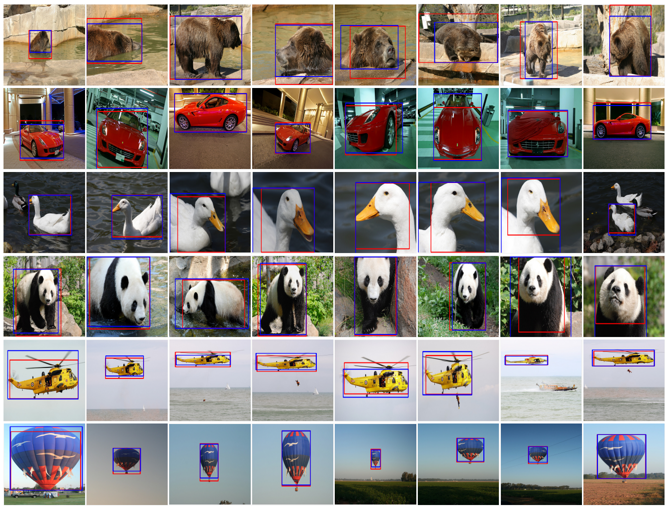


Fig. 10: Co-localization results generated by our approach on the iCoseg dataset. In the six examples (rows), the common object categories are Alaskan brown bear, Ferrari, goose, panda, helicopter, and hot balloon, respectively. In each image, the blue and red bounding boxes represent the ground truth and the estimated results, respectively.

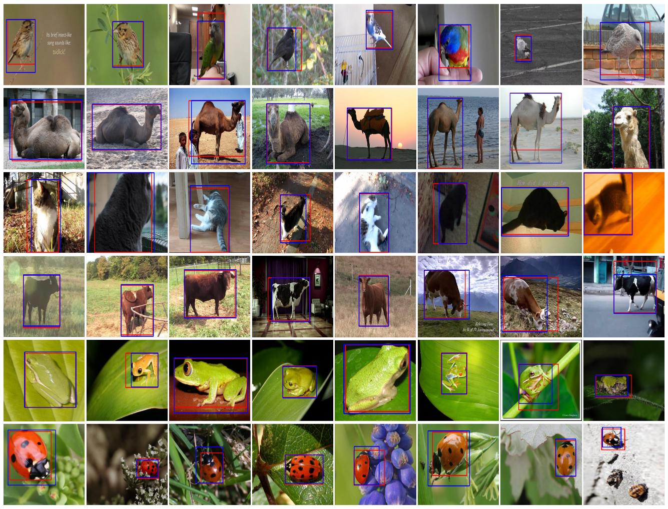


Fig. 11: Co-localization results generated by our approach on the Cosal2015 dataset. In the six examples (rows), the common object categories are bird, camel, cat, cow, frog, and ladybird, respectively. In each image, the blue and red bounding boxes represent the ground truth and the estimated results, respectively.

Compared to the algorithms that require complex optimization processes in CLRW [72] or UODL [73], our method can use a simple thresholding scheme to generate the promising boxes with the aid of the better estimated co-saliency results. Fig. 10 and Fig. 11 give some examples of our results on the iCoseg and Cosal2015 datasets, respectively. Like co-segmentation, object co-localization based on our method is well fulfilled with promising bounding boxes under different types of variations.

VI. Conclusions

We have presented an unsupervised framework for co-saliency detection. Our fusion-learning based model is composed of two stages. First, we propose SAEF to carry out the saliency proposal fusion via jointly exploring the image-level confidence based on the reconstruction error of SAE and the region-level confidence from co-salient object likelihood. Afterwards, our proposed STCNN can gradually learn co-salient objects in a self-taught fashion. The benefits of integrating both the fusion-based and deep-learning-based methods are evident as it produces the co-saliency maps of high quality via making the most of multiple locally complementary saliency proposals. Moreover, unlike existing fusion methods relying on the low-rank assumption of salient foreground regions, we propose a novel idea that takes advantage of the unsupervised SAE into our unified optimization process and generates even better results. In addition to co-saliency detection, our method is applied to two applications—object co-segmentation and object co-localization, in which our method performs favorably against the state-of-the-art methods.

References

[1] H. Fu, D. Xu, S. Lin, and J. Liu, “Object-based RGBD image co-segmentation with mutex constraint,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2015.

[2] K. Jerripothula, J. Cai, and J. Yuan, “Image co-segmentation via saliency co-fusion,” *IEEE Trans. on Multimedia*, 2016.

[3] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang, “Co-attention CNNs for unsupervised object co-segmentation,” in *Proc. Int’l Joint Conf. Artificial Intelligence*, 2018.

[4] K. R. Jerripothula, J. Cai, and J. Yuan, “Quality-guided fusion-based co-saliency estimation for image co-segmentation and co-localization,” *IEEE Trans. on Multimedia*, 2018.

[5] J. Xue, C. Li, and N. Zheng, “Proto-object based rate control for JPEG2000: An approach to content-based scalability,” *IEEE Trans. on Image Processing*, 2011.

[6] C.-C. Tsai, X. Qian, and Y.-Y. Lin, “Image co-saliency detection via locally adaptive saliency map fusion,” in *Proc. Int’l Conf. Acoustics, Speech, and Signal Processing*, 2017.

[7] H. Li and K. Ngan, “A co-saliency model of image pairs,” *IEEE Trans. on Image Processing*, 2011.

[8] H. Fu, X. Cao, and Z. Tu, “Cluster-based co-saliency detection,” *IEEE Trans. on Image Processing*, 2013.

[9] X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng, “Self-adaptively weighted co-saliency detection via rank constraint,” *IEEE Trans. on Image Processing*, 2014.

[10] X. Cao, Y. Cheng, Z. Tao, and H. Fu, "Co-saliency detection via base reconstruction," in *Proc. ACM Int'l Conf. Multimedia*, 2014.

[11] Z. Liu, W. Zou, L. Li, L. Shen, and O. Le Meur, "Co-saliency detection based on hierarchical segmentation," *Signal Processing Letters*, 2014.

[12] L. Ye, Z. Liu, J. Li, W.-L. Zhao, and L. Shen, "Co-saliency detection via co-salient object discovery and recovery," *Signal Processing Letters*, 2015.

[13] Y. Li, K. Fu, Z. Liu, and J. Yang, "Efficient saliency-model-guided visual co-saliency detection," *Signal Processing Letters*, 2015.

[14] D. Zhang, J. Han, J. Han, and L. Shao, "Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining," *IEEE Trans. on Neural Networks and Learning Systems*, 2016.

[15] D. Zhang, J. Han, C. Li, J. Wang, and X. Li, "Detection of co-salient objects by looking deep and wide," *Int'l J. Computer Vision*, 2016.

[16] D. Zhang, D. Meng, and J. Han, "Co-saliency detection via a self-paced multiple-instance learning framework," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2017.

[17] L. Wei, S. Zhao, O. Bourahla, X. Li, and F. Wu, "Group-wise deep co-saliency detection," in *Proc. Int'l Joint Conf. Artificial Intelligence*, 2017.

[18] J. Han, G. Cheng, Z. Li, and D. Zhang, "A unified metric learning-based framework for co-saliency detection," *IEEE Trans. on Circuits and Systems for Video Technology*, 2017.

[19] X. Yao, J. Han, D. Zhang, and F. Nie, "Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering," *IEEE Trans. on Image Processing*, 2017.

[20] C.-C. Tsai, X. Qian, and Y.-Y. Lin, "Segmentation guided local proposal fusion for co-saliency detection," in *Proc. Int'l Conf. Multimedia and Expo*, 2017.

[21] K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, X. Qian, and Y.-Y. Chuang, "Unsupervised cnn-based co-saliency detection with graphical optimization," in *Proc. Euro. Conf. Computer Vision*, 2018.

[22] C.-C. Tsai, W. Li, K.-J. Hsu, X. Qian, and Y.-Y. Lin, "Image co-saliency detection and co-segmentation via progressive joint optimization," *IEEE Trans. on Image Processing*, 2019.

[23] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2015.

[24] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2016.

[25] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2016.

[26] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2017.

[27] P. Zhang, D. Wang, H. Lu, H. Wang, and B. Yin, "Learning uncertain convolutional features for accurate saliency detection," in *Proc. Int'l Conf. Computer Vision*, 2017.

[28] P. Zhang, D. Wang, H. Lu, H. Wang, and X. Ruan, "Amulet: Aggregating multi-level convolutional features for salient object detection," in *Proc. Int'l Conf. Computer Vision*, 2017.

[29] X. Li, F. Yang, H. Cheng, J. Chen, Y. Guo, and L. Chen, "Multi-scale cascade network for salient object detection," in *Proc. ACM Int'l Conf. Multimedia*, 2017.

[30] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Proc. Int'l Conf. Computer Vision*, 2005.

[31] P. Jiang, N. Vasconcelos, and J. Peng, "Generic promotion of diffusion-based salient object detection," in *Proc. Int'l Conf. Computer Vision*, 2015.

[32] W.-C. Tu, S. He, Q. Yang, and S.-Y. Chien, "Real-time salient object detection with a minimum spanning tree," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2016.

[33] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *Proc. Int'l Conf. Computer Vision*, 2017.

[34] X. Li, H. Lu, L. Zhang, X. Ruan, and M.-H. Yang, "Saliency detection via dense and sparse reconstruction," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013.

[35] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013.

[36] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2014.

[37] C. Li, Y. Yuan, W. Cai, Y. Xia, and D. Feng, "Robust saliency detection via regularized random walks ranking," in *CVPR*, 2015.

[38] Y. Tang, X. Wu, and W. Bu, "Saliency detection based on graph-structural agglomerative clustering," in *Proc. ACM Int'l Conf. Multimedia*, 2015.

[39] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 fps," in *Proc. Int'l Conf. Computer Vision*, 2015.

[40] F. Huang, J. Qi, H. Lu, L. Zhang, and X. Ruan, "Salient object detection via multiple instance learning," *IEEE Trans. on Image Processing*, 2017.

[41] K. Fu, I. Gu, and J. Yang, "Saliency detection by fully learning a continuous conditional random field," *IEEE Trans. on Multimedia*, 2017.

[42] Z. Wang, D. Xiang, S. Hou, and F. Wu, "Background-driven salient object detection," *IEEE Trans. on Multimedia*, 2016.

[43] S. Huo, Y. Zhou, J. Lei, N. Ling, and C. Hou, "Iterative feedback control-based salient object segmentation," *IEEE Trans. on Multimedia*, 2018.

[44] C. Aytekin, H. Possegger, T. Mauthner, S. Kiranyaz, H. Bischof, and M. Gabbouj, "Spatiotemporal saliency estimation by spectral foreground detection," *IEEE Trans. on Multimedia*, 2017.

[45] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Neural Information Processing Systems*, 2010.

[46] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. G. Hauptmann, "Self-paced learning for matrix factorization." in *Proc. Nat'l Conf. Artificial Intelligence*, 2015.

[47] L. Jiang, D. Meng, T. Mitamura, and A. Hauptmann, "Easy samples first: Self-paced reranking for zero-example multimedia search," in *Proc. ACM Int'l Conf. Multimedia*, 2014.

[48] J. Supancic and D. Ramanan, "Self-paced learning for long-term tracking," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2013.

[49] D. Gong, M. Tan, Y. Zhang, A. van den Hengel, and Q. Shi, "Self-paced kernel estimation for robust blind image deblurring," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2017.

[50] I. Lillo, J. Niebles, and A. Soto, "A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2016.

[51] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *Int'l J. Computer Vision*, 2004.

[52] J. Pont-Tuset, P. Arbelaez, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2017.

[53] C. Chen, F. Qi, and G. Shi, "Bottom-up visual saliency estimation with deep autoencoder-based sparse reconstruction," *IEEE Trans. on Neural Networks and Learning Systems*, 2016.

[54] M. Grant, S. Boyd, and Y. Ye, "CVX: MATLAB software for disciplined convex programming," 2008.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2016.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int'l Conf. Learning Representations*, 2014.

[57] A. Vedaldi and K. Lenc, "MatConvNet – convolutional neural networks for MATLAB," in *Proc. ACM Int'l Conf. Multimedia*, 2015.

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int'l Conf. Learning Representations*, 2014.

[59] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional models for semantic segmentation," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2015.

[60] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with gaussian edge potentials," in *Proc. Neural Information Processing Systems*, 2011.

[61] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, "iCoseg: Interactive co-segmentation with intelligent scribble guidance," in *Proc. Conf. Computer Vision and Pattern Recognition*, 2010.

[62] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji, "Structure-measure: A new way to evaluate foreground maps," in *Proc. Int'l Conf. Computer Vision*, 2017.

[63] A. Borji, M.-M.-. Cheng, H. Jiang, and J. Li, "Salient object detection: A benchmark," *IEEE Trans. on Image Processing*, 2015.

[64] R. Margolin, L. Zelnik-Manor, and A. Tal, "How to evaluate foreground maps?" in *Proc. Conf. Computer Vision and Pattern Recognition*, 2014.

[65] A. Faktor and M. Irani, "Co-segmentation by composition," in *Proc. Int'l Conf. Computer Vision*, 2013.

[66] H.-S. Chang and Y.-C. F. Wang, "Optimizing the decomposition for multiple foreground cosegmentation," *Computer Vision and Image Understanding*, 2015.

[67] K. Jerripothula, J. Cai, F. Meng, and J. Yuan, “Automatic image co-segmentation using geometric mean saliency,” in *Proc. Int’l Conf. Image Processing*, 2014.

[68] K. Jerripothula, J. Cai, and J. Yuan, “Group saliency propagation for large-scale and quick image co-segmentation,” in *Proc. Int’l Conf. Image Processing*, 2015.

[69] C. Lee, W.-D. Jang, J.-Y. Sim, and C.-S. Kim, “Multiple random walkers and their application to image cosegmentation,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2015.

[70] Z. Tao, H. Liu, H. Fu, and Y. Fu, “Image cosegmentation via saliency-guided constrained clustering with cosine similarity,” in *Proc. Nat’l Conf. Artificial Intelligence*, 2017.

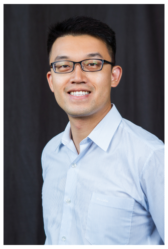
[71] K. Jerripothula, J. Cai, J. Lu, and J. Yuan, “Object co-skeletonization with co-segmentation,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2017.

[72] K. Tang, A. Joulin, L.-J. Li, and F.-F. Li, “Co-localization in real-world images,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2014.

[73] M. Cho, S. Kwak, C. Schmid, and J. Ponce, “Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2015.

[74] X.-S. Wei, C.-L. Zhang, Y. Li, C.-W. Xie, J. Wu, C. Shen, and Z.-H. Zhou, “Deep descriptor transforming for image co-localization,” in *Proc. Int’l Joint Conf. Artificial Intelligence*, 2017.

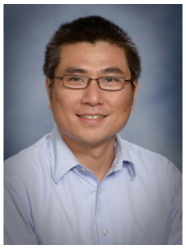
[75] E. Collins, R. Achanta, and S. Susstrunk, “Deep feature factorization for concept discovery,” in *Proc. Euro. Conf. Computer Vision*, 2018.



Chung-Chi “Charles” Tsai received the B.S. degree from National Tsing-Hua University, Hsinchu, Taiwan, and M.S. degree from University of California at Santa Barbara, Santa Barbara, CA, USA, and the Ph.D. degree from Texas A&M University, College Station, TX, USA, in 2009, 2012 and 2018, respectively and all in Electrical Engineering. He attended a one-year exchange program, at the University of New Mexico, Albuquerque, NM, USA, in 2007, and also participated in the summer internship with MediaTek in the summer of 2013/2015/2016. He is currently a senior system engineer for camera ISP in Qualcomm Technologies, Inc, San Diego, CA, USA. His research interests include image processing, computational photography, and computer vision.

Kuang-Jui Hsu received the B.S. degree from the Department of Electrical Engineering, National Sun Yat-sen University in 2011, and the M.S. degree from the Graduate Institute of Networking and Multimedia and Ph.D. degree from the Department of Computer Science and Information Engineering, National Taiwan University, 2013 and 2019, respectively. Currently, he’s a senior computer vision engineer in Qualcomm Taiwan. His research interests include computer vision, machine learning, deep learning, and image processing.

Yen-Yu Lin (M12) received the B.B.A. degree in Information Management, and the M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Taiwan University in 2001, 2003, and 2010, respectively. He is a Professor with the Department of Computer Science, National Chiao Tung University since August 2019. Prior to that, he worked for the Research Center for Information Technology Innovation, Academia Sinica from January 2011 to July 2019. His research interests include computer vision, machine learning, and artificial intelligence.



Xiaoning Qian (S01-M07-SM17) received the Ph.D. degree in Electrical Engineering from Yale University, New Haven, CT, USA. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA. He is affiliated with the TEES-AgriLife Center for Bioinformatics & Genomic Systems Engineering and the Center for Translational Environmental Health Research at Texas A&M University. His recent honors include the National Science Foundation CAREER Award, the Texas A&M Engineering Experiment Station (TEES) Faculty Fellow, and the Montague-Center for Teaching Excellence Scholar at Texas A&M University. His research interests include machine learning and Bayesian computation and their applications in materials science, computational network biology, genomic signal processing, and biomedical signal and image analysis.

Yung-Yu Chuang received his B.S. and M.S. from National Taiwan University in 1993 and 1995 respectively, and the Ph.D. from the University of Washington at Seattle in 2004, all in Computer Science. He is currently a professor with the Department of Computer Science and Information Engineering, National Taiwan University. His research interests include computational photography, computer vision and rendering.