Multi-scale Graph Fusion for Co-saliency Detection

Rongyao Hu[1,2], **Zhenyun Deng**[3], **Xiaofeng Zhu**[1,2,*]

[1]Center for Future Media and School of Computer Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]School of Natural and Computational Science, Massey University Auckland Campus, New Zealand
[3]School of Computer Science, The University of Auckland, New Zealand
hurongyao123, zysjd1991@gmail.com, xfzhu0011@hotmail.com

Abstract

The key challenge of co-saliency detection is to extract discriminative features to distinguish the common salient foregrounds from backgrounds in a group of relevant images. In this paper, we propose a new co-saliency detection framework which includes two strategies to improve the discriminative ability of the features. Specifically, on one hand, we segment each image to semantic superpixel clusters as well as generate different scales/sizes of images for each input image by the VGG-16 model. Different scales capture different patterns of the images. As a result, multi-scale images can capture various patterns among all images by many kinds of perspectives. Second, we propose a new method of Graph Convolutional Network (GCN) to fine-tune the multi-scale features, aiming at capturing the common information among the features from all scales and the private or complementary information for the feature of each scale. Moreover, the proposed GCN method jointly conducts multi-scale feature fine-tune, graph learning, and feature learning in a unified framework. We evaluated our method on three benchmark data sets, compared to state-of-the-art co-saliency detection methods. Experimental results showed that our method outperformed all comparison methods in terms of different evaluation metrics.

Introduction

Co-saliency detection focuses on simulating the human visual system to perceive the scene for searching the common and salient prospects from a group of images (Zhang et al. 2018; Peng et al. 2020), and has been applied to improve the understanding of the image or video content in various applications such as image retrieval (Papushoy and Bors 2015), images co-segmentation (Tsai et al. 2018), and objects co-localization (Jerripothula et al. 2017; Wang et al. 2017). In the co-saliency detection task, the semantic category of the common salient objects should be detected from the specific content of the input image group, involving two key steps, *i.e.,* feature extraction extracting discriminative features to reliably distinguish the foregrounds from the backgrounds of each image, and model construction detecting the co-saliency regions from a group of images based on the extracted features.

Feature extraction is focused on extracting either handcrafted features or deep features based on image pixel or superpixel. The popular methods for handcrafted feature extraction include color/texture feature (Fu, Cao, and Tu 2013; Shen et al. 2018), Histogram of Oriented Gradient (HOG) feature (Huang, Feng, and Sun 2015), GIST descriptors (Jerripothula, Cai, and Yuan 2016), *etc.* Since handcrafted features are usually difficult to capture the appearance changes of both common objects and complex background information (Wang et al. 2019), deep features have been widely designed to explore the semantic connection of co-saliency objects (Tsai et al. 2018; Zhang et al. 2018). For example, (Ren et al. 2020) proposed to extract both deep collaborative features and deep high-to-low features to balance the individual intra-image information. (Wang et al. 2019) employed the VGG-19 framework to extract the high-level group-wise semantic feature and the visual feature for co-saliency detection. Although current feature extraction methods (including handcrafted features and deep features) achieved success in the application of co-saliency detection, extracting single feature is still a challenging task to detect complex variations between co-salient objects and backgrounds. To this end, multi-view feature was extracted to explore both intra-image and inter-image information for co-saliency detection (Jiang et al. 2019a; Zhang et al. 2020a).

Given the image features, both traditional machine learning methods and deep learning methods are designed to detecting the co-saliency across a group of images. For example, (Zhang, Meng, and Han 2016) regarded the co-saliency detection task as multi-instance learning where each image and each superpixel region, respectively, are regarded as a bag and an instance, and thus the multi-instance classifier is used to predict the locations of the co-salient objects in the instance level. However, feature extraction and co-saliency detection are two separated processes in many traditional machine learning methods. As a result, the feature can not be adjusted based on the result of co-saliency detection, and thus leading to suboptimal performance of co-saliency detection. To address this issue, deep learning integrates these two processes in a unified framework so that each other can be adaptively adjusted by the other, and thus easily outputting optimal performance of co-saliency detection. For example, fully convolution neural networks were designed to automatically learn high-level semantic features by mod-

[*]Corresponding author.
Rongyao Hu and Zhenyun Deng contribute equally to this work.

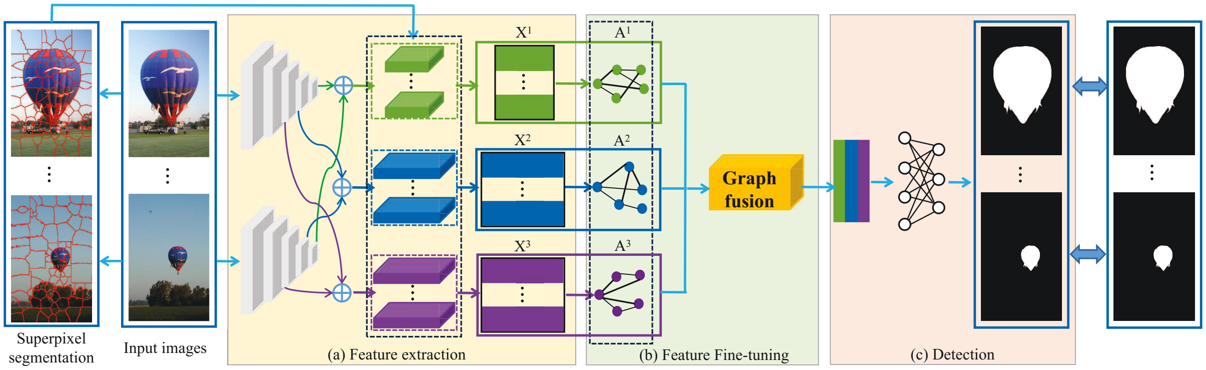


Figure 1: The architecture of the proposed framework for co-saliency detection. Specifically, it involves three key steps, (a) Feature Extraction extracts three-scale deep features to represent each image; (b) Feature Fine-tuning fine-tunes the multi-scale features to obtain discriminative features by considering their common and complementary information; (c) Detection conducts a binary classification task to distinguish the common salient foregrounds from backgrounds.

eling collaborative relationships among the images (Zhang et al. 2019).

Recently, Graph Convolutional Network (GCN) was designed to utilize both the feature information and the relationship among the images (*i.e.,* the graph) to improve the performance of co-saliency detection (Jiang et al. 2019a; Xu et al. 2020). However, previous deep learning methods suffer from some drawbacks to severely limit the detection effectiveness. For example, deep learning methods focus on extracting the self-learnt features from the images without considering the semantic meaning and lacking the interpretability. The convolutional layers and pooling operations in some deep learning methods decrease the size of feature maps to easily result in the loss of boundary details (Zhang et al. 2019).

In this paper, we propose a novel GCN method to fuse multi-scale features based on the superpixel regions/clusters for co-saliency detection. To do this, our proposed method involves three steps, *i.e.,* feature extraction, feature fusion by the proposed GCN method, and co-saliency detection, shown in Figure 1. In the step of feature extraction, we first employ the Simple Linear Iterative Clustering (SLIC) algorithm (Achanta et al. 2012) to obtain superpixel based regions/clusters including sub-blocks of the background and saliency regions for each image. The motivation is that the superpixel representation may adhere to image boundaries better, compared to pixel representation (Zhang et al. 2018). We also employ VGG-16 to generate multi-scale features for each image. Furthermore, we convert multi-scale features to represent the image with a vector based on the superpixel clusters. In the step of feature fine-tuning, we design a new graph fusion method to fine-tune the features of each scale by the help of the information of the features from other scales. The goal is to comprehensively explore the intra-image correlation within one image and the inter-region relationship across the images. Finally, the outputted features are concatenated together first and then passes a fully-connected layer to conduct the binary classification, *i.e.,* regarding the co-saliency detection task as a classification task.

Compared to previous methods, we list the contributions of our method as follows.

- This paper first extracts multi-scale features and then designs a new graph fusion method to fine-tune these features. The multi-scale features can detect different sizes of patterns of the images and the fusion method fine-tunes the multi-scale features to extract the complementary information and the common information among the features. It is noteworthy that previous methods (Zhang et al. 2016; Han et al. 2017) extract handcrafted features to difficult explore the comprehensive information among the images. Other methods extract the multi-view feature to touch the issue of the handcrafted feature, but leaving the correlation among multiple features alone (Liu et al. 2019; Jiang et al. 2019a; Zhang et al. 2020a). Hence, our method is more flexible compared to these methods.
- This paper proposes a new dynamic GCN method jointly conducting multi-graph fusion, graph learning, and feature learning in a unified work. In the literature, (Jiang et al. 2019a) and (Zhang et al. 2020a) focused on conducting multi-graph learning on multi-view data by considering the consistency among the graph (*i.e.,* the common information) and ignoring the complementary or private information across multi-scale features.

Methodology

Overview

Denoting $\mathcal{P} = \{\mathbf{P}_n\}_{n=1}^{N}$ as a set of N related images, co-saliency detection is designed to output the map matrix $\mathcal{M} = \{\mathbf{M}^n\}_{n=1}^{N}$, which is used for distinguishing the common salient foregrounds from backgrounds. To this end, our proposed method includes three steps visualized in Figure 1.

Given the input image set $\mathcal{P}$, we first employ the SLIC algorithm to conduct superpixel segmentation to obtain superpixel regions or clusters. Meanwhile, we employ the VGG-16 model (Simonyan and Zisserman 2014) to convert each input image to multi-scale images by removing the fully-connected layers and the softmax layer of the VGG-16 model. Specifically, we store the images outputted at the third pooling layer, the fourth pooling layer, and the fifth pooling layer. After this, we combine the superpixel regions and the images of each scale to obtain three hierarchical features $\mathcal{X} = \{\mathbf{X}^v\}_{v=1}^3$ as the multi-scale features of $\mathcal{P}$, and thus converting the co-saliency detection task to the classification task based on superpixel clusters.

Feature Extraction

Perceptual and semantic visual features are essential for co-saliency detection (Zha et al. 2020; Zhang et al. 2018). A superpixel is usually defined as a set of pixels with common characteristics such as pixel intensity, so superpixel based handcrafted features were shown to carry more semantic information and contain perceptual meaning, compared to either pixel based handcrafted features (Gao et al. 2020). However, handcrafted features are not robust to complex visual scenes (Zhang et al. 2020b; Ren et al. 2020). On the contrary, deep features can capture the changes within one image or across images to produce robust co-saliency detection models, but lacking semantic meaning. In this paper, we propose to integrate superpixel segmentation with deep features to generate multi-scale deep features for each image, aiming at producing semantic and discriminative features as well as converting the co-saliency detection task to a classification problem based on the superpixel cluster/region.

Given a set of N related images $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^N$ ($\mathbf{P}_i \in \mathbb{R}^{224\times224\times3}$, we employ the SLIC algorithm (Achanta et al. 2012) to generate superpixel regions for each image $\mathbf{P}_i$ by clustering pixels based on their color similarity and proximity in the image plane. As a result, we obtain n_i superpixels for each image. For simplicity, we set all n_is ($i = 1, ..., N$) as the same value for a group of images, *i.e.,* n and denote $\mathcal{N} = N \times n$. Meanwhile, we input each image $\mathbf{P}_i$ to the pre-trained VGG-16 model to generate three images with different scales, *i.e.,* $\mathbf{P}_i \rightarrow \{\tilde{\mathbf{P}}_i^1, \tilde{\mathbf{P}}_i^2, \tilde{\mathbf{P}}_i^3\}$ where $\tilde{\mathbf{P}}_i^1 \in \mathbb{R}^{56\times56\times256}$, $\tilde{\mathbf{P}}_i^2 \in \mathbb{R}^{28\times28\times512}$, and $\tilde{\mathbf{P}}_i^3 \in \mathbb{R}^{14\times14\times512}$, respectively, denotes the images obtained from the third pooling layer, the fourth pooling layer, and the fifth pooling layer of the VGG-16 model. We then upsampling these images to be the equivalent size, *i.e.,* $\tilde{\mathbf{X}}_i^1 \in \mathbb{R}^{224\times224\times256}$, $\tilde{\mathbf{X}}_i^2 \in \mathbb{R}^{224\times224\times512}$, and $\tilde{\mathbf{X}}_i^3 \in \mathbb{R}^{224\times224\times512}$ where 256 and 512 indicate the filter number, aiming to avoid the loss of boundary details due to the decrease of the feature map size in $\{\tilde{\mathbf{P}}_i^1, \tilde{\mathbf{P}}_i^2, \tilde{\mathbf{P}}_i^3\}$ (Gao et al. 2020).

For each image $\tilde{\mathbf{X}}_i^j$ ($i = 1, ..., N$ and $j = 1, 2, 3$), we use the result of the superpixel segmentation to partition it into n regions. The representation of each superpixel region is a scalar, which is the average values of the activation maps of all pixels within the same superpixel. Hence, each image is represented by three matrices with different scales of the image size, *e.g.,* $\mathbf{X}_i^1 \in \mathbb{R}^{n\times256}$, $\mathbf{X}_i^2 \in \mathbb{R}^{n\times512}$, and $\mathbf{X}_i^3 \in \mathbb{R}^{n\times512}$, $i = 1, ..., N$. Furthermore, we use $\mathcal{X} = \{\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3\}$ to represent the feature matrices of relevant images, where $\mathbf{X}^1 \in \mathbb{R}^{\mathcal{N}\times256}$, $\mathbf{X}^2 \in \mathbb{R}^{\mathcal{N}\times512}$, and $\mathbf{X}^3 \in \mathbb{R}^{\mathcal{N}\times512}$. Finally, the initial graph matrix $\mathbf{A}^v$ ($v = 1, 2, 3$) for $\mathbf{X}^v$ is constructed by the formulation: $\mathbf{A}^v = \mathbf{X}^v\mathbf{X}^{vT} \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$ (Jiang et al. 2019a).

Feature Fine-tuning

In this section, we first review the classical GCN model and then propose our proposed graph fusion method in details.

Graph convolutional network The GCN method aims to learn a latent representation $\mathbf{O}^v = f(\mathbf{X}^v, \mathcal{G}^v; \Theta^v)$ of the original feature matrix $\mathbf{X}^v$ ($v = 1, 2, 3$) while preserving the graph structure of all data points (Kipf and Welling 2016). Generally, GCN includes one input layer, two hidden layers, and one perceptron layer. Given the input matrix $\mathbf{X}^v \in \mathbb{R}^{\mathcal{N}\times d_v}$ which has $\mathcal{N}$ superpixel regions (or samples) and d_v features for each sample, $\mathbf{A}^v$ denotes the pair-wise correlation between any two samples. Hence, the layer-wise propagation in the k-th hidden layer of GCN is

$$\mathbf{F}_{k+1}^v = \sigma(\tilde{\mathbf{D}}^{v^{-\frac{1}{2}}}\tilde{\mathbf{A}}^v\tilde{\mathbf{D}}^{v^{-\frac{1}{2}}}\mathbf{F}_k^v\mathbf{\Theta}_k^v) \qquad (1)$$

where $k = (0, 1, ..., K-1)$ and K is the number of layers. $\mathbf{F}_0^v = \mathbf{X}^v$ is the initial feature matrix, $\mathbf{F}_k^v$ is the output feature map of the k-th layer, $\tilde{\mathbf{A}}^v = \mathbf{A}^v + \mathbf{I}_n$ is the adjacency matrix of the undirected graph, and $\mathbf{I}_n$ is the identity matrix. $\tilde{\mathbf{D}}^v = diag(\tilde{\mathbf{d}}_1^v, ..., \tilde{\mathbf{d}}_n^v)$ is a diagonal matrix with $\tilde{\mathbf{d}}_i^v = \sum_j^n \tilde{\mathbf{A}}^v$ and $\sigma(.)$ is an activation function such as ReLU. The last perception layer is defined as:

$$\mathbf{O}^v = softmax(\tilde{\mathbf{D}}^{v^{-\frac{1}{2}}}\tilde{\mathbf{A}}^v\tilde{\mathbf{D}}^{v^{-\frac{1}{2}}}\mathbf{F}_K^v\mathbf{\Theta}_K^v) \qquad (2)$$

$\mathbf{O}^v$ is the prediction matrix, $\mathbf{\Theta}^v = (\mathbf{\Theta}_0^v, ..., \mathbf{\Theta}_K{}^v)$ which are trainable parameters and can be learned by minimizing the cross-entropy loss function over the labeled samples.

$$\mathcal{L}_{GCN} : -\sum_{i\in L}\sum_{j}^{c} y_{ij} ln o_{ij}^v \qquad (3)$$

where L denotes the set of labelled samples, c is the number of classes, y_{ij} is the ground truth, and o_{ij}^v is the corresponding predictions.

Different from Convolutional Neural Network (CNN) (Krizhevsky, Sutskever, and Hinton 2012) regrading the feature matrix as the input, GCN regards both the feature matrix and the graph as the inputs to generate deep features by preserving the local structure in the graph. As a result, GCN has been demonstrated to outperform CNN in many real applications (Kipf and Welling 2016). Moreover, previous studies (*e.g.,* (Chen, Wu, and Zaki 2019; Jiang et al. 2019b)) showed that the quality of the graph is the key issue for the effectiveness of the GCN method. In the literature, many methods can be used for constructing the graph, *i.e.,* k Nearest Neighbor (kNN) graph, ϵ-neighborhood graph, fully connected graph, *etc.* The graph construction by many previous GCN methods is independent of the feature learning process, so that easily resulting in the sub-optimal feature learning. To address this issue, dynamic GCN methods focus on jointly conducting graph learning and feature learning, where the graph can be updated by the optimal features and the features are also adjusted by the update graph. As a result, the quality of the graph can be improved by a data-driven way, and thus the outputted feature is discriminative. To this end, the following objective function of the graph learning is:

$$\begin{aligned}\mathcal{L}_{GL} : \min_{\mathbf{A}^v} & \textstyle\sum_{i,j=1}^n \|\mathbf{x}_i^v\mathbf{Q}^v - \mathbf{x}_j^v\mathbf{Q}^v\|_2^2 a_{ij}^v + \|\mathbf{A}^v\|_F^2 \\ & s.t., \textstyle\sum_{j=1}^n a_{ij}^v = 1, a_{ij}^v > 0, i, j = 1, ..., n.\end{aligned} \qquad (4)$$

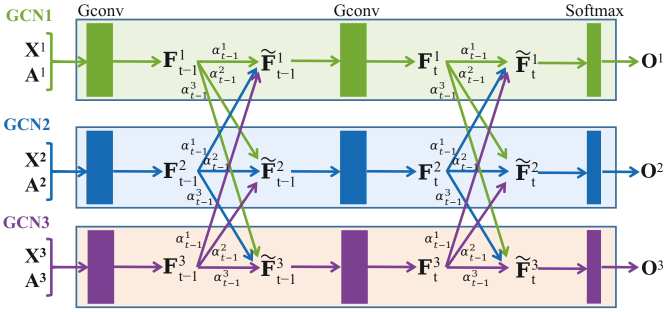


Figure 2: The structure of the proposed graph fusion, which conducts feature fine-tuning by exploring the common and complementary information of multi-scale features.

where $\mathbf{Q}^v \in \mathbb{R}^{d_v \times r}$ ($r \leq d_v$) and a^v_{ij} denote the similarity between $\mathbf{x}^v_i$ and $\mathbf{x}^v_j$. Finally, the dynamic GCN method adds the constraint of the graph learning (*i.e.,* Eq. (4)) as the regularization of the GCN model to have:

$$\mathcal{L} = \mathcal{L}_{GCN} + \gamma \mathcal{L}_{GL} \tag{5}$$

where γ is a tuning parameter.

Similar to the literature (Li et al. 2018; Jiang et al. 2019a) that approximately optimizes a new variable with less tuning parameters rather than directly optimizing the variable $\mathbf{Q}^v$ in Eq. (4) with expensive time cost, this paper designs to optimize $\mathbf{Q}^v$ by the following objective function:

$$\mathbf{A}^v = \sigma(\mathbf{X}^v \mathbf{Q}^v (\mathbf{X}^v \mathbf{Q})^T) \tag{6}$$

where $\sigma(.)$ is the sigmoid activation function and $\mathbf{Q}^v$ is learnable projection matrix.

Proposed graph fusion In this work, we design a new dynamic GCN in Eq. (5) and Eq. (6) to fine-tune the features of each scale, which was obtained from VGG and superpixel segmentation. Thus we obtain a dynamic GCN model for the features from each scale. However, each GCN model is independently trained from other two. Hence, we propose a fusion method to combine three dynamic GCN models to explore the common information among three models and the complementary information in each model. We list the proposed fusion structure in Figure 2.

Specifically, given the feature matrix $\mathbf{X}^v$ and the corresponding graph $\mathbf{A}^v$, the layer-wise propagation in the hidden layer of our proposed GCN method is defined as:

$$\mathbf{F}^v_t = ReLU(\hat{\mathbf{A}}^v \hat{\mathbf{F}}^v_{(t-1)} \mathbf{\Theta}^v_t) \tag{7}$$

where $\mathbf{F}^v_t \in \mathbb{R}^{\mathcal{N} \times d^t_v}$ is the new representation of $\hat{\mathbf{F}}^v_{(t-1)}$ in the t-th layer, $\hat{\mathbf{A}}^v = \mathbf{D}^{v^{-\frac{1}{2}}}(\mathbf{A}^v + \mathbf{I}_{\mathcal{N}})\mathbf{D}^{v^{-\frac{1}{2}}}$ is normalized adjacency matrix, and $\mathbf{D}^v$ is the diagonal matrix of $(\mathbf{A}^v + \mathbf{I}_{\mathcal{N}})$. $\mathbf{I}_{\mathcal{N}}$ is an identity matrix and $ReLU(.)$ is an activation function. $\mathbf{\Theta}^v_t$ is a trainable projection matrix for the v-th superpixel feature.

Since we have multi-scale features to describe the same patterns on a group of images. The features with different sizes/scales can capture the common foregrounds with different scales. Moreover, the features of each scale has the complementary information (or private information, *e.g.,* different foregrounds) different from the features from other scales, while all features should have the common information (*i.e.,* the common foregrounds with different scales) as they are assumed to contain the same foregrounds. If the common information is detected, these features will be discriminative for the co-saliency detection. Meanwhile, the difference among the features can also benefit the learning of discriminative features. To this end, we have the definition of $\tilde{\mathbf{F}}^v_{(t-1)}$ as follows:

$$\tilde{\mathbf{F}}^v_{(t-1)} = \sum_{v=1}^{V} \alpha^v_{(t-1)} \mathbf{F}^v_{(t-1)} \tag{8}$$

where $\alpha^v_{(t-1)}$ indicates the contribution or the weight for $\mathbf{F}^v_{(t-1)}$ to its v-th final features $\tilde{\mathbf{F}}^v_{(t-1)}$ in the (t-1)-th layer. Moreover, $\boldsymbol{\alpha}^v_{(t-1)} = [\alpha^1_{(t-1)}, ..., \alpha^V_{(t-1)}]$ is a trainable vector. Specifically, our GCN method outputs $\mathbf{F}^v_{(t-1)}$, which will be combined with all other $\mathbf{F}^{v'}_{(t-1)}$ ($v \neq v'$) to generate the v-th final features $\tilde{\mathbf{F}}^v_{(t-1)}$ in the (t-1)-th layer. As a result, the feature learning in each scale have the complementary information (*i.e.,* $\mathbf{F}^v_{(t-1)}$) and the common information from other scales $\tilde{\mathbf{F}}^{v'}_{(t-1)}$ ($v \neq v'$).

After the new presentation $\mathbf{F}^v_t$ is obtained, its final output is defined as:

$$\mathbf{O}^v = softmax(\tilde{\mathbf{A}}^v \tilde{\mathbf{F}}^v_t \mathbf{\Theta}^v_t) \tag{9}$$

After conducting Eq. (9), we obtain three outputs and then concatenate them to have:

$$\mathbf{Z} = FC([\mathbf{O}^1, \mathbf{O}^2, \mathbf{O}^3]) \tag{10}$$

where FC denotes the fully-connected layer and $\mathbf{Z}$ is the predicted label.

Co-saliency detection is designed to propagate information from intra-superpixel correlations across the relevant images. Hence, we only consider the prediction performance by employing the cross-entropy loss function to obtain:

$$\begin{aligned} \mathcal{L}_{cos} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{n} \eta^i (\mathbf{z}^i(j) log \mathbf{z}^i(j) \\ - (1 - \eta^i)(1 - \mathbf{z}^i(j)) log(1 - \mathbf{y}^i(j))) \end{aligned} \tag{11}$$

where $\mathbf{y}^i(j)$ and $\mathbf{z}^i(j)$ is the ground truth and predicted result of the j-th superpixel of the i-th image, respectively. η^i is the ratio of salient superpixel cluster in all superpixel clusters and can be calculated by applying the same superpixel partition for ground truths in advance.

Experiments

We experimentally evaluated our method, compared to four comparison methods, on three image data sets, *i.e.,* iCoseg, Cosal2015, and MSRC, in terms of four evaluation metrics.

Data Sets

The data set iCoseg (Batra et al. 2010) contains 643 images within 38 different categories. Each image has a manually labeled pixel-wise ground truth for evaluation.

Methods	iCoseg				Cosal2015				MSRC			
	AUC ↑	F_β ↑	S_α ↑	AP ↑	AUC ↑	F_β ↑	S_α ↑	AP ↑	AUC ↑	F_β ↑	S_α ↑	AP ↑
CBCS	0.9315	0.7301	0.6707	0.7958	0.8077	0.5489	0.5439	0.5859	0.8083	0.6563	0.4959	0.6992
ESMG	0.9317	0.7094	0.7436	0.7728	0.7687	0.4803	0.5524	0.5111	0.7875	0.6111	0.5452	0.6112
EGNet	0.9598	0.8651	0.8365	0.8751	0.9303	0.7909	0.8206	0.8077	0.8624	0.7714	0.7183	0.7618
MGLCN	0.9671	**0.8912**	0.8355	0.8263	0.9534	0.8845	0.8142	0.8519	0.9415	0.8559	0.8001	0.8427
Proposed	**0.9727**	0.8787	**0.8391**	0.8742	**0.9716**	**0.8928**	**0.9341**	**0.8817**	**0.9515**	**0.8565**	**0.8212**	**0.9158**

Table 1: Results of all methods on three image data sets.

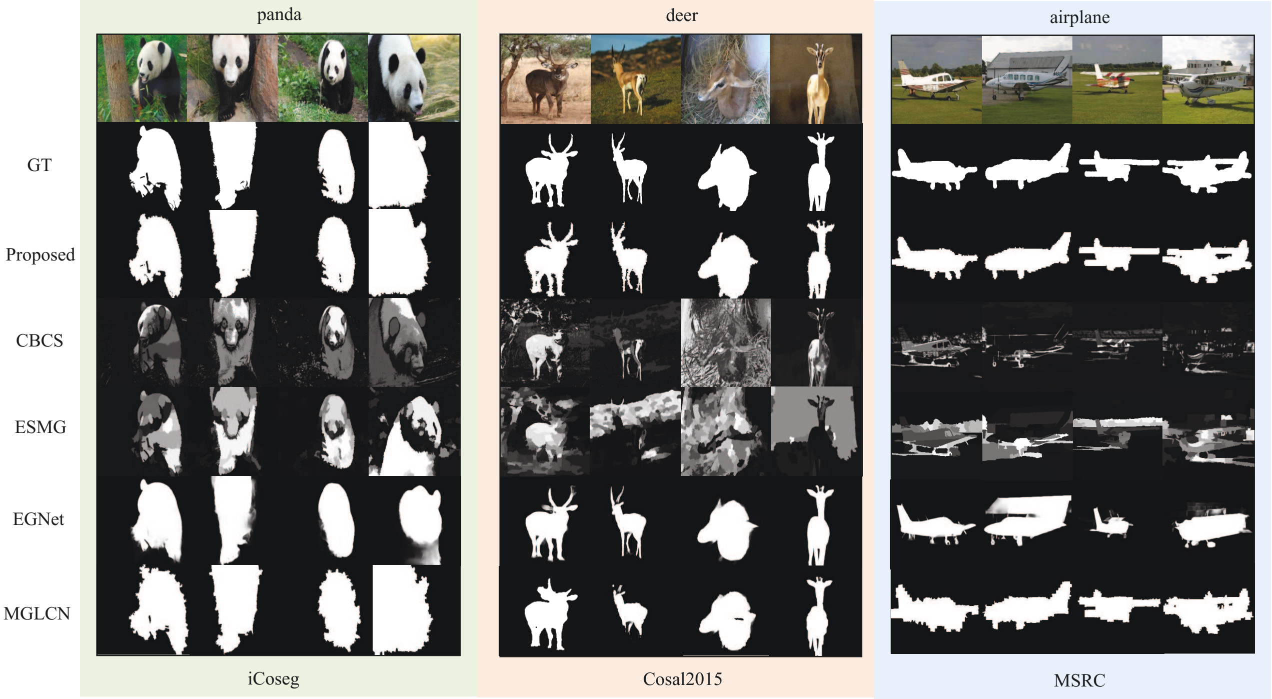


Figure 3: Visualization comparisons of all methods on three images, each of which is from one data set.

The data set Cosal2015 (Zhang et al. 2016) consists of 2015 images of 50 categories, and each group suffers from various challenging issues such as complex environments, occlusion issues, target appearance variations, and background clutters.

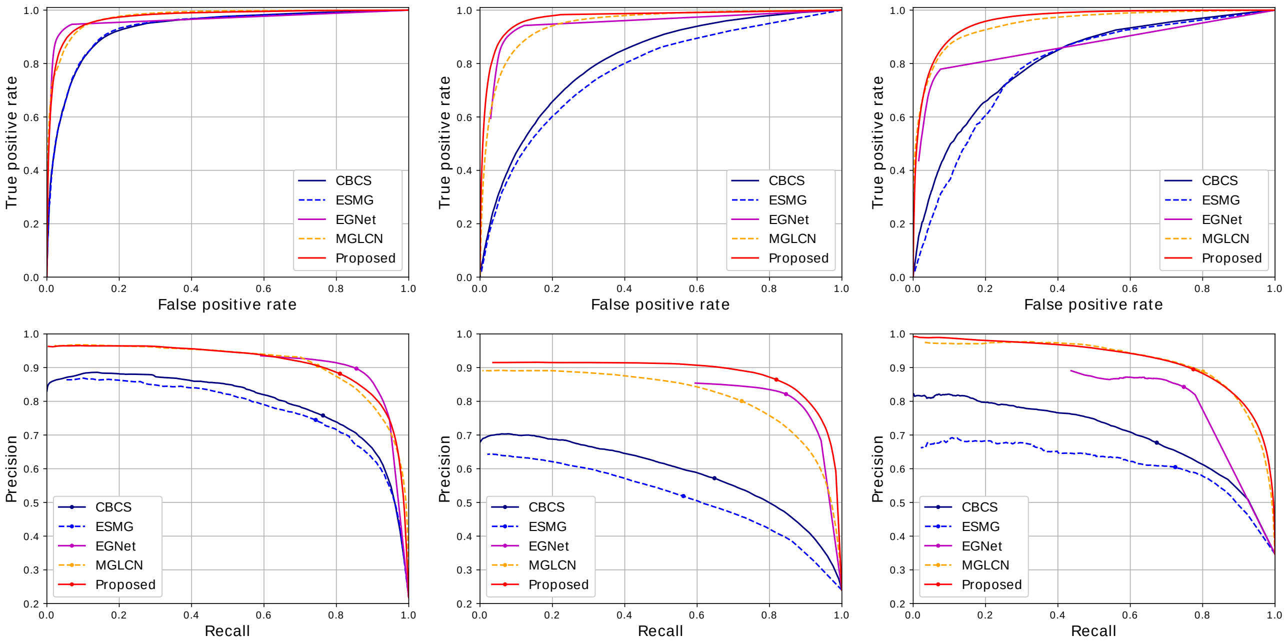
The data set MSRC (Winn, Criminisi, and Minka 2005) contains 233 images within 7 categories. The images in the data set are complicated as the common objects are vary unpredictable in color and shape appearance.

Comparison Methods

We used four state-of-the-art methods of co-saliency detection to evaluate the effectiveness of our proposed framework in our experiments.

- Cluster-Based Co-Saliency detection (**CBCS**) integrates three bottom-up saliency cues (including the spatial distribution cue, the global contrast cue, and the corresponding cue) with multiplication way to conduct the final co-saliency maps (Fu, Cao, and Tu 2013).
- Efficient Saliency-Model-Guided co-saliency detection (**ESMG**) conducts a two-step saliency-guided method, where the first step uses the manifold ranking to recover the co-salient parts missing for each single saliency map and the second step utilizes a ranking framework with various queries to capture the corresponding correlations to guide co-saliency maps (Li et al. 2014).
- Edge Guidance Network (**EGNet**) designs a single base network which consists of three parts, *i.e.,* edge feature extraction, salient object feature extraction, and one-to-one guidance network, to improve the saliency detection performance (Zhao et al. 2019).
- Multiple Graph Learning and Convolutional Network (**MGLCN**) explores the superpixel-level similarity to replace pixel-level saliency detection, by embedding both the intra-graph and the inter-graph learning in the framework of graph convolution network (Jiang et al. 2019a).

The methods (*e.g.,* CBCS and ESMG) are traditional machine learning methods and the methods (*e.g.,* EGNet, MGLCN, and our method) are deep learning methods.



(a) iCoseg

(b) Cosal2015

(c) MSRC

Figure 4: ROC and PR curves of all methods on three image data sets.

Experiment Setting

In our experiments, we reshaped the size of all images to 224×224 and set the number of superpixel regions as 5000. For deep learning methods (*i.e.,* EGNet, MGLCN, and our method), we selected the data set MSRAB in (Liu et al. 2010) to train deep models. In our method, we set the maximal number of epochs as 10000 using the Adam optimizer (Kingma and Ba 2014), and set the initial learning rate and the weight decay, respectively, as 1e-5 and 0.005. We set stopping criterion as no decreasing of the objective function for 100 consecutive epochs in the training process. For fair comparison, we obtained the source codes by online or from the authors. The experimental settings of all comparison methods were followed the corresponding literature to make all of them output their best performance. All experiments were conducted on a server with 4 NVIDIA Quadro P4000 8G.

The evaluation metrics included Precision-Recall (PR) curve, Receiver Operating Characteristic (ROC) curve, Area Under the Curve (AUC) score , F_β score, S_α score, and Average Precision (AP) (Fan et al. 2017). Specifically, F_β score is defined as:

$$F_\beta = \frac{(1+\beta^2)Precision \times Recall}{\beta^2 Precision + Recall} \tag{12}$$

where precision and recall are obtained using a self-adaptive threshold $T = \mu + \varepsilon$. μ and ε are the mean and standard deviation values of the saliency map, respectively. We followed (Achanta et al. 2009) to set β^2 as 0.3.

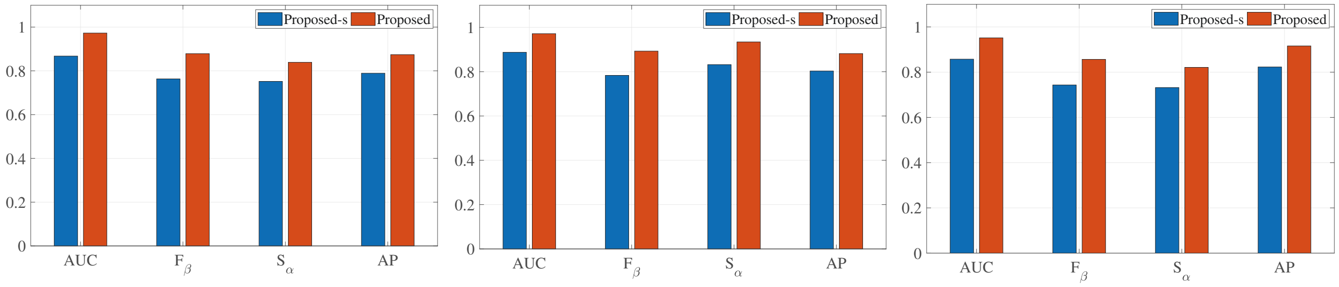
S_α score describes the structural similarity between the ground truths and the corresponding co-saliency maps, and we followed the literature (Fan et al. 2017) to set all hyperparameters as 0.5.

Results Analysis

We listed the results of all methods on three benchmark data sets in Table 1, where the bold number stands for the best result in one column. We also reported the ROC and PR curves of all methods on all data sets in Figure 4.

First, our proposed framework obtained the best performance, followed by MGLCN, EGNet, CBCS, and ESMG. For example, our method improved on average by 1.13%, 0.12%, 4.82%, and 5.03%, compared to the best comparison method (*i.e.,* MGLCN), and averagely improved by 13.60%, 27.57%, 25.11%, and 25.89%, compared to the worst comparison method (*i.e.,* ESMG), in terms of AUC, F_β, S_α, and AP, respectively, on three data sets. This indicates the success of our two strategies for co-saliency detection, *i.e.,* generating multi-scale images for every image, and fusing multi-scale features to produce discriminative features. In particular, deep learning methods (*i.e.,* EGNet, MGLCN, and our method) outperformed traditional methods (*i.e.,* CBCS and ESMG) as the former methods extract more informative features to describe the salient region than the latter ones. This indicates that deep features are suitable for co-saliency detection.

Second, by comparing with four deep learning methods, EGNet achieved the worst performance as the methods (such as MGLCN, and our method) extract multiple deep features for co-saliency detection. For example, MGLCN improved on average by 3.65%, 6.81%, 2.48%, and 2.54%, respectively, on three data sets, for the evaluation metrics such as AUC, F_β, S_α, and AP, compared to EGNet. This implies that graph convolutional structure are reasonable for co-saliency detection.



(a) iCoseg

(b) Cosal2015

(c) MSRC

Figure 5: Results of our model without/with the process of feature fusion on three data sets.

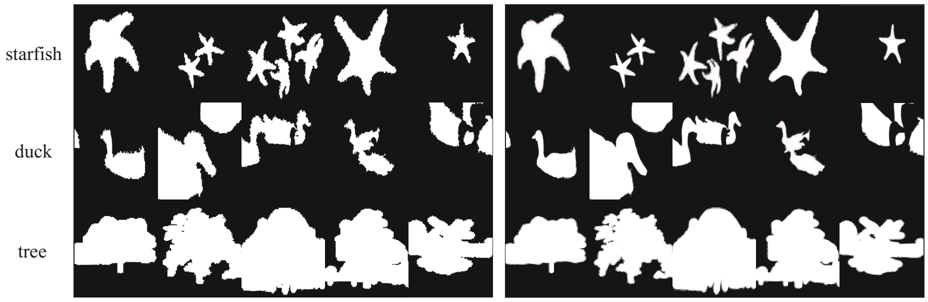


Figure 6: Visual comparisons between the classification task (left) and the regression task (right) using our framework on three images for the used data sets.

Ablation Analysis

In this section, we verify the effectiveness of our model from the following aspects: (1) the effectiveness of our fusion method; and (2) regression performance of our method.

Graph fusion effectiveness In our framework, we fuse the features from multiple scales to explore the complementary information in each scale and the common information among all scales. However, we can also ignore the fusion process, *i.e.,* separately conducting 3 dynamic models and then concatenate 3 outputs to conduct co-saliency detection, Proposed-s for short. We reported the results of both Proposed and Proposed-s in Figure 5.

Obviously, Proposed outperformed Proposed-s on all data sets in terms of different evaluation metrics. For example, Proposed improved by on average 9.4%, 11.32%, 8.93%, and 9.29%, respectively, compared to Proposed-s, in terms of AUC, F_β, S_α, and AP. This indicates the importance for feature fusion on multi-scale features.

Regression effectiveness In this paper, we regarded the co-saliency detection task as a binary classification task, and reported the visualization of all methods in Figure 3. Actually, we can also regard the co-saliency detection task as a regression task, whose visualization can easier detect the edge boundary compared to the classification task. This is because that the regression task assigns the edge boundary with continuous values and the classification task assigns it with binary values. To this end, we reported the visualization of our method on the regression task in Figure 6.

Compared the regression task to the classification task in terms of the visualization, the edge boundary produced by the regression task is more blur by considering the pixel graph-scale values, compared to the one in the classification task. Hence, the proposed framework can be designed for both the classification task and the regression task.

Conclusion

In this paper, we proposed a new co-saliency detection framework by designing two strategies to generate discriminative features, *i.e.,* multi-scale features to capture the patterns with different sizes across the images, and feature fusion to extract the common and complementary information among the multi-scale features. Moreover, we embedded these two strategies into our designed dynamic GCN model to jointly conduct feature fusion, graph learning, and feature learning. Experimental results on three benchmark data sets demonstrated that our framework outperformed the state-of-the-art methods of co-saliency detection in terms of several evaluation metrics. Moreover, experimental results also verified the effectiveness of each strategy in our co-saliency detection framework.

Acknowledgements

This work was partially supported by the Natural Science Foundation of China (Grants No: 61876046 and 61672177); the Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing; the Guangxi Bagui Teams for Innovation and Research; the Marsden Fund of New Zealand (MAU1721); the Project of Guangxi Science and Technology (GuiKeAD17195062); and the Sichuan Science and Technology Program (No. 2019YFG0533).

References

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

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence* 34(11): 2274–2282.

Batra, D.; Kowdle, A.; Parikh, D.; Luo, J.; and Chen, T. 2010. i-coseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 3169–3176.

Chen, Y.; Wu, L.; and Zaki, M. J. 2019. Deep iterative and adaptive learning for graph neural networks. *arXiv preprint arXiv:1912.07832* .

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

Fu, H.; Cao, X.; and Tu, Z. 2013. Cluster-based co-saliency detection. *IEEE Transactions on Image Processing* 22(10): 3766–3778.

Gao, G.; Zhao, W.; Liu, Q.; and Wang, Y. 2020. Co-Saliency Detection with Co-Attention Fully Convolutional Network. *IEEE Transactions on Circuits and Systems for Video Technology* .

Han, J.; Quan, R.; Zhang, D.; and Nie, F. 2017. Robust object co-segmentation using background prior. *IEEE Transactions on Image Processing* 27(4): 1639–1651.

Huang, R.; Feng, W.; and Sun, J. 2015. Saliency and co-saliency detection by low-rank multiscale fusion. In *ICME*, 1–6.

Jerripothula, K. R.; Cai, J.; Lu, J.; and Yuan, J. 2017. Object co-skeletonization with co-segmentation. In *CVPR*, 3881–3889.

Jerripothula, K. R.; Cai, J.; and Yuan, J. 2016. Cats: Co-saliency activated tracklet selection for video co-localization. In *ECCV*, 187–202.

Jiang, B.; Jiang, X.; Zhou, A.; Tang, J.; and Luo, B. 2019a. A unified multiple graph learning and convolutional network model for co-saliency estimation. In *ACM MM*, 1375–1382.

Jiang, B.; Zhang, Z.; Lin, D.; Tang, J.; and Luo, B. 2019b. Semi-supervised learning with graph learning-convolutional networks. In *CVPR*, 11313–11320.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* .

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Li, R.; Wang, S.; Zhu, F.; and Huang, J. 2018. Adaptive graph convolutional neural networks. *AAAI* .

Li, Y.; Fu, K.; Liu, Z.; and Yang, J. 2014. Efficient saliency-model-guided visual co-saliency detection. *IEEE Signal Processing Letters* 22(5): 588–592.

Liu, T.; Yuan, Z.; Sun, J.; Wang, J.; Zheng, N.; Tang, X.; and Shum, H.-Y. 2010. Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence* 33(2): 353–367.

Liu, Y.; Han, J.; Zhang, Q.; and Shan, C. 2019. Deep salient object detection with contextual information guidance. *IEEE Transactions on Image Processing* 29: 360–374.

Papushoy, A.; and Bors, A. G. 2015. Image retrieval based on query by saliency content. *Digital Signal Processing* 36: 156–173.

Peng, L.; Yang, Y.; Wang, Z.; Huang, Z.; and Shen, H. T. 2020. MRA-Net: Improving VQA via Multi-modal Relation Attention Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10.1109/TPAMI.2020.3004830.

Ren, J.; Liu, Z.; Li, G.; Zhou, X.; Bai, C.; and Sun, G. 2020. Co-Saliency Detection Using Collaborative Feature Extraction And High-To-Low Feature Integration. In *ICME*, 1–6.

Shen, F.; Xu, Y.; Liu, L.; Yang, Y.; Huang, Z.; and Shen, H. T. 2018. Unsupervised Deep Hashing with Similarity-Adaptive and Discrete Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40(12): 3034–3044.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Tsai, C.-C.; Li, W.; Hsu, K.-J.; Qian, X.; and Lin, Y.-Y. 2018. Image co-saliency detection and co-segmentation via progressive joint optimization. *IEEE Transactions on Image Processing* 28(1): 56–71.

Wang, B.; Yang, Y.; Xu, X.; Hanjalic, A.; and Shen, H. T. 2017. Adversarial Cross-Modal Retrieval. In *ACM MM*, 154–162.

Wang, C.; Zha, Z.-J.; Liu, D.; and Xie, H. 2019. Robust deep co-saliency detection with group semantic. In *AAAI*, volume 33, 8917–8924.

Winn, J.; Criminisi, A.; and Minka, T. 2005. Object categorization by learned universal visual dictionary. In *ICCV*, volume 2, 1800–1807.

Xu, X.; Wang, T.; Yang, Y.; Hanjalic, A.; and Shen, H. T. 2020. Radial Graph Convolutional Network for Visual Question Generation. *IEEE Transactions on Neural Networks and Learning Systems* 10.1109/TNNLS.2020.2986029.

Zha, Z.-J.; Wang, C.; Liu, D.; Xie, H.; and Zhang, Y. 2020. Robust Deep Co-Saliency Detection With Group Semantic and Pyramid Attention. *IEEE Transactions on Neural Networks and Learning Systems* .

Zhang, D.; Fu, H.; Han, J.; Borji, A.; and Li, X. 2018. A review of co-saliency detection algorithms: Fundamentals, applications, and challenges. *ACM Transactions on Intelligent Systems and Technology* 9(4): 1–31.

Zhang, D.; Han, J.; Li, C.; Wang, J.; and Li, X. 2016. Detection of co-salient objects by looking deep and wide. *International Journal of Computer Vision* 120(2): 215–232.

Zhang, D.; Meng, D.; and Han, J. 2016. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(5): 865–878.

Zhang, K.; Li, T.; Liu, B.; and Liu, Q. 2019. Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing. In *CVPR*, 3095–3104.

Zhang, K.; Li, T.; Shen, S.; Liu, B.; Chen, J.; and Liu, Q. 2020a. Adaptive Graph Convolutional Network with Attention Graph Clustering for Co-saliency Detection. In *CVPR*, 9050–9059.

Zhang, Z.; Jin, W.; Xu, J.; and Cheng, M.-M. 2020b. Gradient-Induced Co-Saliency Detection. *arXiv preprint arXiv:2004.13364* .

Zhao, J.-X.; Liu, J.-J.; Fan, D.-P.; Cao, Y.; Yang, J.; and Cheng, M.-M. 2019. EGNet: Edge guidance network for salient object detection. In *ICCV*, 8779–8788.