CVPR
#2579

CVPR
#2579

CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Adaptive Graph Convolutional Network with Attention Graph Clustering for Co-saliency Detection

Anonymous CVPR submission

Paper ID 2579

Abstract

Co-saliency detection aims to discover the common and salient foregrounds from a group of relevant images. For this task, we present a novel adaptive graph convolutional network with attention graph clustering (GCAGC). Three major contributions have been made, and are experimentally shown to have substantial practical merits. First, we propose a graph convolutional network design to extract information cues to characterize the intra- and inter-image correspondence. Second, we develop an attention graph clustering algorithm to discriminate the common objects from all the salient foreground objects in an unsupervised fashion. Third, we present a unified framework with encoder-decoder structure to jointly train and optimize the graph convolutional network, attention graph cluster, and co-saliency detection decoder in an end-to-end manner. We evaluate our proposed GCAGC method on three co-saliency detection benchmark datasets (iCoseg, Cosal2015 and COCO-SEG). Our GCAGC method obtains significant improvements over the state-of-the-arts on most of them. Code will be made available.

1. Introduction

Human is able to exhibit visual fixation to attend to the attractive and interesting regions and objects for future processing [7]. Co-saliency detection model simulates the human visual system to perceive the scene, and searches for the common and salient foregrounds in an image group. Co-saliency has been used in various applications to improve the understanding of image/video content, such as image/video co-segmentation [55, 13, 14, 59], object co-localization [25, 53], and image retrieval [47, 68].

In co-saliency detection, the semantic categories of the common salient objects are unknown. Thus, the designed algorithm needs to infer such information from the specific content of the input image group. Therefore, the co-saliency detection algorithm design usually focuses on addressing t-
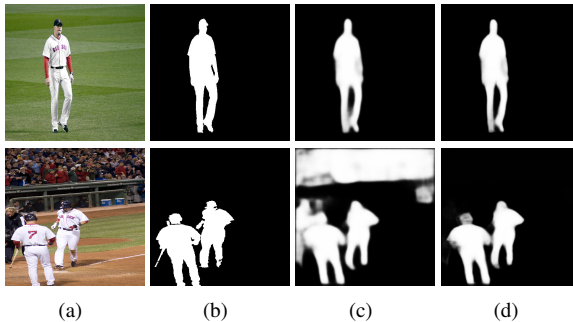

Figure 1. Illustration of our different results. (a) Input images; (b) Ground truth; (c) Corresponding maps only with designed adaptive graph convolutional network; (d) Final co-saliency maps with our proposed GCAGC.

wo key challenges: (1) extracting informative image feature representations to robustly describe the image foregrounds; and (2) designing effective computational frameworks to formulate and detect the co-saliency. Conventional hand-engineered features, such as Gabor filters, color histograms and SIFT descriptors [43] have been widely used in many co-saliency detection methods [12, 41, 69]. However, hand-crafted shallow features usually lack the ability to fully capture the large variations of common object appearances, and complicated background textures [57]. Recently, researchers improve co-saliency detection using deep-learning-based high-level feature representations, and have shown promising results [75, 73, 76]. Nonetheless, these approaches separate the representation extraction from co-saliency detection as two distinct steps, and lose the ability to tailor the image features towards inferring co-salient regions [24]. End-to-end algorithms adopting convolutional neural networks (CNNs) [24, 57, 64] have been developed to overcome this problem, and demonstrated state-of-the-art performance. Although CNN is able to extract image representations in a data-driven way, it is the sub-optimal solution to model long-range dependencies [61]. CNN captures long-range dependencies by deeply stacking convolutional operations to enlarge the receptive fields. However, the

CVPR
#2579

CVPR
#2579

CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

repeated convolutional operations cause optimization difficulties [61, 21], and make multi-hop dependency modeling [61]. Moreover, it becomes even more challenging for the CNN to accurately modeling the inter-image non-local dependencies for the co-salient regions in the image group.

To address the aforementioned challenges, we develop a novel adaptive graph convolutional network with attention graph clustering (GCAGC) for co-saliency detection. We first utilize a CNN encoder to extract multi-scale feature representations from the image group, and generate combined dense feature node graphs. We then process the dense graphs with the proposed adaptive graph convolutional network (AGCN). Compared with only depending on the progressive behavior of the CNN, the AGCN is able to capture the non-local and long-range correspondence directly by computing the interactions between any two positions of the image group, regardless of their intra- and inter-image positional distance. The output from AGCN is further refined by an attention graph clustering module (AGCM) through generated co-attention maps. A CNN decoder is employed in the end to output the finally predicted co-saliency maps. A unified framework is designed to jointly optimize all the components together. Figure 1 illustrates the effectiveness of the proposed methods.

The main contributions of this paper are threefold:

- We provide an adaptive graph convolutional network design to simultaneously capture the intra- and inter-image correspondence of an image group. Compared with conventional approaches, this AGCN directly computes the long-range interactions between any two image positions, thus providing more accurate measurements.

- We develop an attention graph clustering module to differentiate the common objects from salient foregrounds. This AGCM is trained in an unsupervised fashion, and generates co-attention maps to further refine the estimated co-salient foregrounds.

- We present an end-to-end computational framework with encoder-decoder CNN structure to jointly optimize the graph clustering task and the co-saliency detection objective, while learning adaptive graph dependencies.

2. Related Work

Image Co-saliency Detection. This task identifies common distinct foregrounds and segments them from multiple images. Various strategies have been developed for this task. Bottom-up approaches first score each pixel/sub-region in the image group, and then combine similar regions in a bottom-up fashion. Hand-crafted features [12, 15, 35, 41, 69] or deep-learning-based features [75, 74] are usually employed to score such sub-regions. Fu et al. [12] utilize three visual attention priors in a cluster-based framework. Liu et al. [41] define background and foreground cues to capture the intra- and inter-image similarities. Pretrained CNN and restricted Boltzmann machine are used in [75] and [74] to extract information cues to detect common salient objects, respectively. In contrast, fusion-based algorithms [54, 5, 26] are designed to discover useful information from the predicted results generated by several existing saliency or co-saliency detection methods. These methods fuse the detected region proposals by region-wise adaptive fusion [26], adaptive weight fusion [5] or stacked-autoencoder-enabled fusion [54]. Learning-based methods are the third category of co-saliency detection algorithms, and developed to learn the co-salient pattern directly from the image group. In [24], an unsupervised CNN with two graph-based losses is proposed to learn the intra-image saliency and cross-image concurrency, respectively. Zhang et al. [76] design a hierarchical framework to capture co-salient area in a mask-guided fully CNN. Wei et al. [64] design a multi-branch architecture to discover the interaction across images and the salient region in single image simultaneously. A semantic guided feature aggregation architecture is proposed to capture the concurrent and fine-grained information in [57]. Although many methods have been developed, this field still lacks of research on addressing the limitations of CNN for capturing long-range intra- and inter-image dependencies.

Graph Neural Networks (GNNs). GNNs [18, 49] are the models for capturing graph dependencies via message passing between the nodes of graphs. Different from standard neural network, GNNs retain a state that can represent information from its neighborhood with arbitrary depth [78]. Convolutional graph neural networks (GCNs) [4, 8, 29, 31, 1, 45, 16] are a variant of GNNs, and aim to generalize convolution to graph domain. Algorithms in this direction are often categorized as the spectral-based approaches [4, 8, 29, 31], and the spatial-based approaches [1, 45, 16]. The former ones work with a spectral representation of the graphs; and the latter ones define the operation directly on graph, and extract information from groups of spatially connected neighbours. Recently, GNN and GCN have demonstrated promising results in various computer vision tasks, including scene graph generation [67, 36, 19], point clouds classification and segmentation [30, 63], semantic segmentation [58, 48], action recognition [66] and visual reasoning and question answering [6, 44]. More comprehensive review of GNNs can be found in [78, 65].

Graph Clustering. This task divides the graph nodes into related groups. Early works [17, 52] develop shallow approaches for graph clustering. Girvan et al. [17] use centrality indices to discover boundaries of different nodes groups.

CVPR
#2579

CVPR
#2579

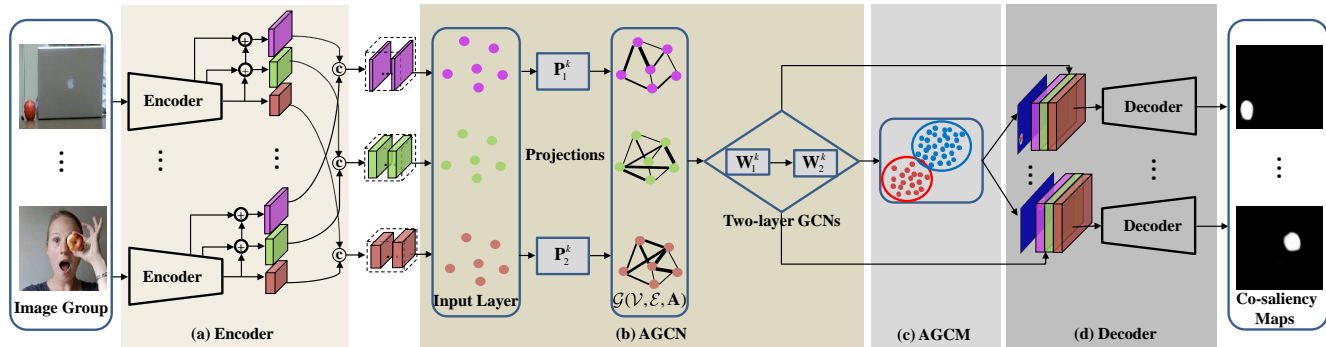CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 2. Pipeline of the proposed GCAGC for co-saliency detection. Given a group of images as input, we first leverage a backbone CNN as encoder (a) to extract the multi-scale features of each image, and then we adopt the feature pyramid network (FPN) [38] to fuse all the image features from top to down. Next, the lateral output features as node representations are fed into the AGCN (b). The output features of AGCN via two-layer GCNs are then fed into the AGCM (c), generating a set of object co-attention maps. Finally, the co-attention maps and the output features of AGCN are concatenated and fed into the decoder (d), producing corresponding co-saliency maps. $\oplus$: element-wise addition; $\copyright$: concatenation; $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{A})$: graph of nodes $\mathcal{V}$, edges $\mathcal{E}$ and adjacency matrix $\mathbf{A}$; $\mathbf{P}_1^k, \mathbf{P}_2^k$: learnable projection matrices for graph learning; $\mathbf{W}_1^k$ and $\mathbf{W}_2^k$: learnable weight matrices in the adopted two-layer GCNs.

Wang *et al.* [60] develop a modularized non-negative matrix factorization approach to incorporate the community structure into the graph embedding, and then perform traditional clustering methods on the embedded features. The limitations of these works are that they only handle partial graph structure or shallow relationships between the content and the structure data [56]. In contrast, deep-learning-based approaches [46, 56] are developed recently to improve graph clustering. Pan *et al.* [46] present an adversarially regularized framework to extract the graph representation to perform graph clustering. Wang *et al.* [56] develop a goal-directed deep learning approach to jointly learn graph embedding and graph clustering together. More detailed review of graph clustering is provided in [2].

3. Proposed Approach

3.1. Method Overview

Given a group of N relevant images $\mathcal{I} = \{\boldsymbol{I}^n\}_{n=1}^N$, the task of co-saliency detection aims to highlight the shared salient foregrounds against backgrounds, predicting the corresponding response maps $\mathcal{M} = \{\mathbf{M}^n\}_{n=1}^N$. To achieve this goal, we learn a deep GCAGC model to predict $\mathcal{M}$ in an end-to-end fashion.

Figure 2 illustrates the pipeline of our approach, which consists of four key components: (a) Encoder, (b) AGC-N, (c) AGCM and (d) Decoder. Specifically, given input $\mathcal{I}$, we first adopt the VGG16 backbone network [51] as the encoder to extract their features by removing the fully-connected layers and softmax layer. Afterwards, we leverage the FPN [38] to fuse the features of pool3, pool4 and pool5 layers, generating three lateral intermediate feature maps $\mathcal{X} = \{\mathbf{X}^k\}_{k=1}^3$ as the multi-scale feature rep-

resentations of $\mathcal{I}$, Then, for each $\mathbf{X}^k \in \mathcal{X}$, we design a sub-graph $\mathcal{G}^k$ with a learnable structure that is adaptive to our co-saliency detection task, which is able to well capture the long-range intra- and inter-image correspondence while preserving the spatial consistency of the saliency. Meanwhile, to fully capture multi-scale information for feature enhancement, the sub-graphs are combined into a multi-graph $\mathcal{G} = \cup_k \mathcal{G}^k$. Then, $\mathcal{G}$ is integrated into a simple two-layer GCNs $\mathcal{F}_{gcn}$ [29], generating the projected GC filtered features $\mathcal{F}_{gcn}(\mathcal{X}) = \{\mathcal{F}_{gcn}(\mathbf{X}^k)\}_{k=1}^3$. Recent works [32, 33] show that the GC filtering of GC-Ns [29] is a Laplacian smoothing process, and hence it makes the salient foreground features of the same category similar, thereby well preserving spatial consistency of the foreground saliency, which facilitates the subsequent intra- and inter-image correspondence. Afterwards, $\mathcal{F}_{gcn}(\mathcal{X})$ are fed into a graph clustering module $\mathcal{F}_{gcm}$, producing a group of co-attention maps $\boldsymbol{M}_{catt}$, which help to further refine the predicted co-salient foregrounds while suppressing the noisy backgrounds. Finally, the concatenated features $\boldsymbol{M}_{catt} \copyright \mathcal{F}_{gcn}(\mathcal{X})$ are fed into a decoder layer, producing the finally predicted co-saliency maps.

3.2. Adaptive Graph Convolution Network

As aforementioned, the AGCN is to process features as Laplacian smoothing [32] that can benefit long-range intra- and inter-image correspondence while preserving spatial consistency. Numerous graph based works for co-saliency detection [24, 55, 77, 23, 27, 37] have been developed to better preserve spatial consistency, but they perform intra-saliency detection and inter-image correspondence independently, which cannot well capture the interactions between co-salient regions across images that are essential to
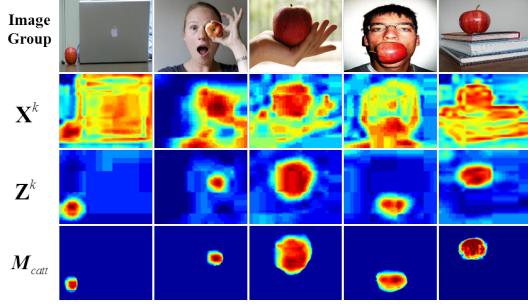
CVPR
#2579

CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

CVPR
#2579

Figure 3. Illustration of the effect of GC filtering. The GC filtered signal projections $\mathbf{Z}^k$ preserve better spatial consistency of the salient foregrounds than the input graph signals $\mathbf{X}^k$ that highlight more noisy backgrounds. Afterwards, the co-attention maps $\boldsymbol{M}_{catt}$ generated by our AGCM in § 3.3 further reduce the noisy backgrounds existing in $\mathbf{Z}^k$.

co-saliency detection, thereby leading to sub-optimal performance. Differently, our AGCN constructs a dense graph that takes all input image features as the node representations. Meanwhile, each edge of the graph models the interactions between any pair-wise nodes regardless of their positional distance, thereby well capturing long-range dependencies. Hence, both intra-saliency detection and inter-image correspondence can be jointly implemented via feature propagation on the graph under a unified framework without any poster-processing, leading to a more accurate co-saliency estimation than those individually processing each part [24, 55, 77, 23, 27, 37].

Notations of Graph. We construct a multi-graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{A}) = \cup_{k=1}^3 \mathcal{G}^k(\mathcal{V}^k, \mathcal{E}^k, \mathbf{A}^k)$ that is composed of three sub-graphs $\mathcal{G}^k$, where node set $\mathcal{V} = \{\mathcal{V}^k\}$, edge set $\mathcal{E} = \{\mathcal{E}^k\}$, adjacent matrix $\mathbf{A} = \sum_k \mathbf{A}^k$, $\mathcal{V}^k = \{v_i^k\}$ denotes the node set of $\mathcal{G}^k$ with node v_i^k, $\mathcal{E}^k = \{e_{ij}^k\}$ denotes its edge set with edge e_{ij}^k, $\mathbf{A}^k$ denotes its adjacent matrix, whose entry $\mathbf{A}^k(i,j)$ denotes the weight of edge e_{ij}^k. $\mathbf{X}^k = [\boldsymbol{x}_1^k, \ldots, \boldsymbol{x}_{Nwh}^k]^\top$ denotes the feature matrix of $\mathcal{G}^k$, where $\boldsymbol{x}_i^k \in \mathbb{R}^{d^k}$ is the features of node v_i^k with dimension d^k.

Adjacency Matrix A. The vanilla GCNs [29] construct a fixed graph without training, which cannot guarantee to be best suitable to a specific task [22]. Recently, some works [22, 34, 27] have investigated adaptive graph learning techniques through learning a parameterized adjacency matrix tailored to a specific task. Inspired by this and the self-attention mechanism in [61], for sub-graph k, to learn a task-specific graph structure, we define a learnable adjacency matrix as

$$\mathbf{A}^k = \sigma(\mathbf{X}^k \mathbf{P}_1^k (\mathbf{X}^k \mathbf{P}_2^k)^\top), \qquad (1)$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ denotes the sigmoid function, $\mathbf{P}_1^k, \mathbf{P}_2^k \in \mathbb{R}^{d^k \times r}$ are two learnable projection matrices
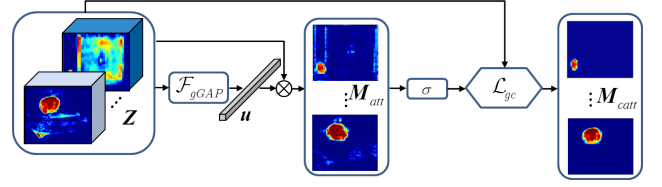
that reduce the dimension of the node features from d^k to $r < d^k$.

To combine multiple graphs in GCNs, as in [62], we simply element-wisely add the adjacency matrices of all $\mathcal{G}^k$ to construct the adjacency matrix of $\mathcal{G}$ as

$$\mathbf{A} = \mathbf{A}^1 + \mathbf{A}^2 + \mathbf{A}^3. \qquad (2)$$

Graph Convolutional Filtering. We employ the two-layer GCNs proposed by [29] to perform graph convolutions as

$$\begin{aligned} \mathbf{Z}^k &= \mathcal{F}_{gcn}(\mathbf{X}^k) \\ &= \mathcal{F}_{softmax}(\hat{\mathbf{A}} \mathrm{ReLU}(\mathcal{F}_{gcf}(\hat{\mathbf{A}}, \mathbf{X}^k) \mathbf{W}_1^k) \mathbf{W}_2^k), \end{aligned} \qquad (3)$$

where the GC filtering function is defined as [33]

$$\mathcal{F}_{gcf}(\hat{\mathbf{A}}, \mathbf{X}^k) = \hat{\mathbf{A}} \mathbf{X}^k, \qquad (4)$$

$\mathbf{W}_1^k \in \mathbb{R}^{d^k \times c_1^k}$, $\mathbf{W}_2^k \in \mathbb{R}^{c_1^k \times c^k}$ denote the learnable weight matrices of two fully-connected layers for feature projections, $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}}$, where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\mathbf{A}$ is defined by (2) and $\mathbf{I}$ denotes the identity matrix, $\tilde{\mathbf{D}}(i,i) = \sum_j \tilde{\mathbf{A}}(i,j)$ is the degree matrix of $\tilde{\mathbf{A}}$ that is diagonal.

Recent work [33] has shown that the GC filtering $\mathcal{F}_{gcf}$ (4) is low-pass and hence it can make the output signal projections $\mathbf{Z}^k$ smoother in the same cluster, so as to well preserve the spatial consistency of the salient foregrounds across images as illustrated by Figure 3. However, some intra-consistency but non-salient regions have also been highlighted. To overcome this issue, in the following section, we will present an attention graph clustering technique to further refine $\mathbf{Z}^k$ to focus on co-salient regions.

3.3. Attention Graph Clustering Module

Figure 4 shows the schematic diagram of our AGCM $\mathcal{F}_{gcm}$. Specifically, given the GC filtering projections $\mathbf{Z}^k \in \mathbb{R}^{Nwh \times c^k}$, $k = 1, 2, 3$ in (3), we obtain a multi-scale feature matrix by concatenating them as $\mathbf{Z} = [\mathbf{Z}^1, \mathbf{Z}^2, \mathbf{Z}^3] = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{Nwh}]^\top \in \mathbb{R}^{Nwh \times d}$, where the multi-scale node features $\boldsymbol{z}_i \in \mathbb{R}^d$, $d = \sum_k c^k$. Next, we reshape $\mathbf{Z}$ to tensor $\mathbf{Z} \in \mathbb{R}^{N \times w \times h \times d}$ as input of $\mathcal{F}_{gcm}$. Then, we define a group global average pooling (gGAP) function $\mathcal{F}_{gGAP}$ as

$$\boldsymbol{u} = \mathcal{F}_{gGAP}(\mathbf{Z}) = \frac{1}{Nwh} \sum_{n,i,j} \mathbf{Z}(n,i,j,:), \qquad (5)$$

Figure 4. The schematic diagram of our AGCM $\mathcal{F}_{gcm}$. Please refer to the text part for details.

CVPR
#2579

CVPR
#2579

CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

which outputs a global statistic feature $\boldsymbol{u} \in \mathbb{R}^d$ as the multi-scale semantic saliency representation that encodes the global useful group-wise context information. Afterwards, we correlate $\boldsymbol{u}$ and $\boldsymbol{Z}$ to generate a group of attention maps that can fully highlight the intra-saliency:

$$\boldsymbol{M}_{att} = \boldsymbol{u} \otimes \boldsymbol{Z}, \tag{6}$$

where $\boldsymbol{M}_{att} \in \mathbb{R}^{N \times w \times h}$, $\otimes$ denotes correlation operator. Then, we use sigmoid function σ to re-scale the values of $\boldsymbol{M}_{att}$ to $[0, 1]$ as

$$\boldsymbol{W} = \sigma(\boldsymbol{M}_{att}). \tag{7}$$

From Figure 4, we can observe that $\boldsymbol{M}_{att}$ discovers intra-saliency that preserves spatial consistency, but some noisy non-co-salient foregrounds have also been highlighted. To alleviate this issue, we exploit an attention graph clustering technique to further refine the attention maps, which are able to better differentiate the common objects from salient foregrounds. Motivated by the weighted kernel k-means approach in [10], we define the objective function of AGCM as

$$\mathcal{L}_{gc} = \sum_{z_i \in \pi_f} w_i \| \boldsymbol{z}_i - \boldsymbol{m}_f \|^2 + \sum_{z_i \in \pi_b} w_i \| \boldsymbol{z}_i - \boldsymbol{m}_b \|^2, \tag{8}$$

where π_f and π_b denote the clusters of foreground and background respectively, $\boldsymbol{m}_f = \frac{\sum_{z_i \in \pi_f} z_i w_i}{\sum_{z_i \in \pi_f} w_i}$ and similar for $\boldsymbol{m}_b$, w_i denotes the i-th element of $\boldsymbol{W}$ in (7).

Following [10], we can readily show that the minimization of the objective $\mathcal{L}_{gc}$ in (8) is equivalent to

$$\min_{\mathbf{Y}} \{ \mathcal{L}_{gc} = -\text{trace}(\mathbf{Y}^\top \mathbf{K} \mathbf{Y}) \}, \tag{9}$$

where $\mathbf{K} = \mathbf{D}^{\frac{1}{2}} \mathbf{Z} \mathbf{Z}^\top \mathbf{D}^{\frac{1}{2}}$, $\mathbf{D} = \text{diag}(w_1, \ldots, w_{Nwh})$, $\mathbf{Y} \in \mathbb{R}^{Nwh \times 2}$ satisfies $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}$.

Let $\boldsymbol{y} \in \{0, 1\}^{Nwh}$ denote the indictor vector of the clusters, and $\boldsymbol{y}(i) = 1$ if $i \in \pi_f$, else, $\boldsymbol{y}(i) = 0$. We choose $\mathbf{Y} = [\boldsymbol{y}/\sqrt{|\pi_f|}, (\mathbf{1} - \boldsymbol{y})/\sqrt{|\pi_b|}]$ that satisfies $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}$ and put it into (9), yielding the loss function of our AGCM

$$\mathcal{L}_{gc} = -\left(\frac{\boldsymbol{y}^\top \mathbf{K} \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} + \frac{(\mathbf{1} - \boldsymbol{y})^\top \mathbf{K} (\mathbf{1} - \boldsymbol{y})}{(\mathbf{1} - \boldsymbol{y})^\top (\mathbf{1} - \boldsymbol{y})} \right). \tag{10}$$

Now, we show the relationship between the above loss $\mathcal{L}_{gc}$ and graph clustering. We first construct the graph of GC as $\mathcal{G}_{gc}(\mathcal{V}_{gc}, \mathcal{E}_{gc}, \mathbf{K})$, which is made up of node set $\mathcal{V}_{gc} = \mathcal{V}_f \cup \mathcal{V}_b$, where $\mathcal{V}_f$ is the set of foreground nodes and $\mathcal{V}_b$ is the set of background nodes, $\mathcal{E}_{gc}$ denotes the edge set such that the weight of edge between nodes i and j is equal to $\mathbf{K}(i, j)$, where $\mathbf{K}$ is its adjacency matrix defined in (9). Let us denote $\text{links}(\mathcal{V}_l, \mathcal{V}_l) = \sum_{i \in \mathcal{V}_l, j \in \mathcal{V}_l} \mathbf{K}(i, j), l = f, b$, then, it is easy to show that minimizing $\mathcal{L}_{gc}$ (10) is equivalent to maximizing the ratio association objective [50] for graph clustering task

$$\max \left\{ \sum_{l=f,g} \frac{\text{links}(\mathcal{V}_l, \mathcal{V}_l)}{|\mathcal{V}_l|} \right\}. \tag{11}$$

where $|\mathcal{V}_l|$ denotes the cardinality of set $\mathcal{V}_l$.

Directly optimizing $\mathcal{L}_{gc}$ (10) yields its continuous relaxed solution $\hat{\boldsymbol{y}}$. Then, we reshape $\hat{\boldsymbol{y}}$ into a group of N co-attention maps $\boldsymbol{M}_{catt} \in \mathbb{R}^{N \times w \times h}$. Finally, the learned co-attention maps $\boldsymbol{M}_{catt}$ and the input features $\boldsymbol{Z} \in \mathbb{R}^{N \times w \times h \times d}$ of the AGCM are concatenated, yielding the enhanced features $\boldsymbol{F} \in \mathbb{R}^{N \times w \times h \times (d+1)}$:

$$\boldsymbol{F} = \boldsymbol{M}_{catt} \copyright \boldsymbol{Z}, \tag{12}$$

where $\copyright$ denotes concatenation operator, which serves as the input of the following decoder network.

3.4. Decoder Network

Our decoder network has an up-sampling module that is consist of a 3×3 convolutional layer to decrease feature channels, a ReLU layer and a deconvolutional layer with stride = 2 to enlarge resolution. Then, we repeat this module three times until reaching the finest resolution for accurate co-saliency map estimation, following a 1×1 convolutional layer and a sigmoid layer to produce a group of co-saliency map estimations.

Given the features $\boldsymbol{F}$ computed by (12) as input, the decoder network generates a group of co-saliency maps $\mathcal{M} = \{\mathbf{M}^n \in \mathbb{R}^{w \times h}\}_{n=1}^N$. We then leverage a weighted cross-entropy loss for pixel-wise classification

$$\mathcal{L}_{cls} = -\frac{1}{P \times N} \sum_{n=1}^N \sum_{i=1}^P \{ \rho^n \mathbf{M}^n(i) \log(\mathbf{M}_{gt}^n(i)) \\ -(1 - \rho^n)(1 - \mathbf{M}^n(i)) \log(1 - \mathbf{M}_{gt}^n(i)) \}, \tag{13}$$

where $\mathbf{M}_{gt}^n$ denotes the ground-truth mask of image $\boldsymbol{I}^n \in \mathcal{I}$, P denotes the pixel number of image $\boldsymbol{I}^n$ and ρ^n denotes the ratio of all positive pixels over all pixels in image $\boldsymbol{I}^n$.

All the network parameters are jointly learned by minimizing the following multi-task loss function

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{gc}, \tag{14}$$

where $\mathcal{L}_{gc}$ is the attention graph clustering loss defined by (10), $\lambda > 0$ is a trade-off parameter. We train our network by minimizing $\mathcal{L}$ in an end-to-end manner, and the learned GCAGC model is directly applied to processing input image group, predicting the corresponding co-saliency maps without any post-processing.

4. Results and Analysis

4.1. Implementation Details

The training of our GCAGC model includes two stages:

CVPR
#2579

CVPR
#2579

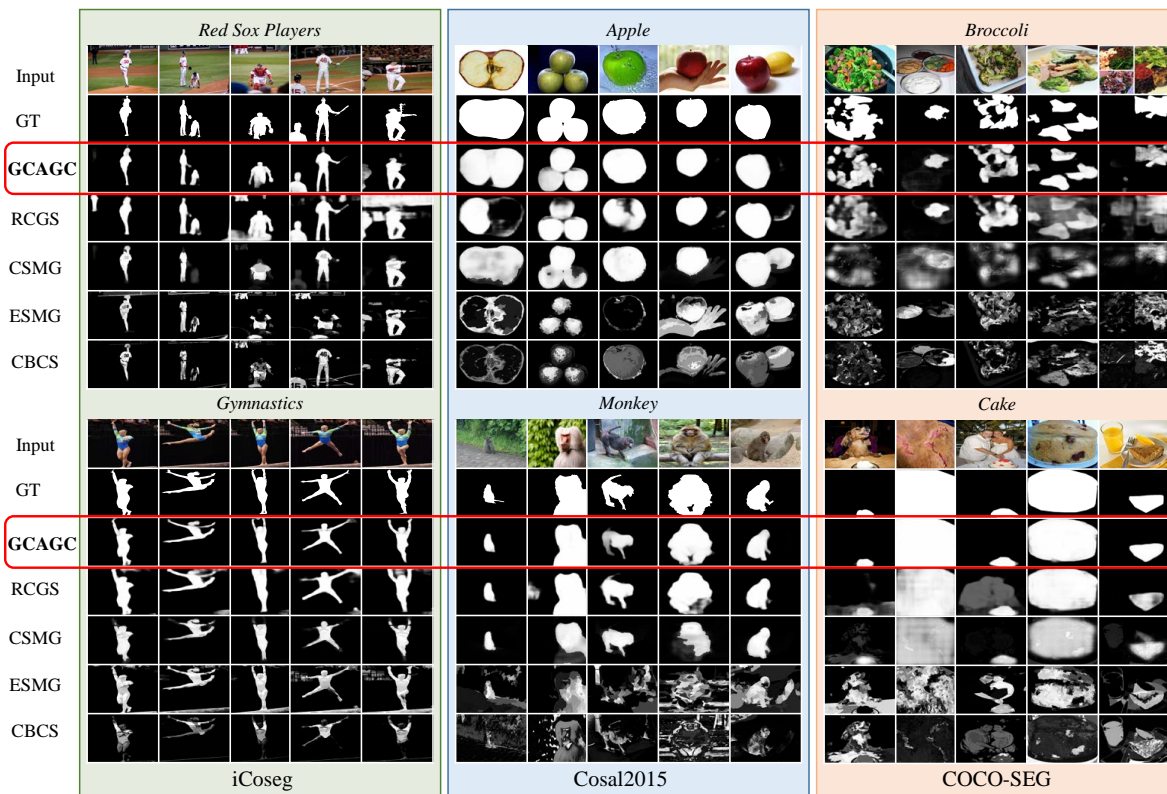CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 5. Visual comparisons of our GCAGC method compared with other state-of-the-arts, including CBCS [12], ESMG [35], CSMG [76] and RCGS [57].

Stage 1. For fair comparison, we adopt the VGG16 network [51] as the backbone network, which is pre-trained on the ImageNet classification task [9]. Following the input settings in [64, 57], we randomly select $N = 5$ images as one group from one category and then select a mini-batch groups from all categories in the COCO dataset [39], which are sent into the network at the same time during training. All the images are resized to the same size of 224×224 for easy processing. The model is optimized by the Adam algorithm [28] with a weight decay of 5e-4 and an initial learning rate of 1e-4 which is reduced by a half every $25,000$ iterations. This training process converges until $100,000$ iterations.

Stage 2. We further fine-tune our model using MSRA-B dataset [40] to better focus on the salient areas. All the parameter settings are the same as those in **Stage 1** except for the training iterations $=10,000$. Note that when training, to match the size of input group, we augment the single salient image to $N = 5$ different images as a group using affine transformation, horizontal flipping and left-right flipping.

During testing, we divide all images into several mini-groups to produce the final co-saliency map estimations. The network is implemented in PyTorch with a RTX 2080Ti GPU for acceleration.

4.2. Datasets and Evaluation Metrics

We conduct extensive evaluations on three popular datasets including iCoseg [3], Cosal2015 [72] and COCO-SEG [57]. Among them, iCoseg is the most widely used dataset with totally 38 groups of 643 images, among which the common objects in one group share similar appearance or semantical characteristics, but have various pose or color changes. Cosal2015 is a large-scale dataset which is consist of $2,015$ images of 50 categories, and each group suffers from various challenging factors such as complex environments, occlusion issues, target appearance variations and background clutters, *etc*. All these increase the difficulty for accurate co-saliency detection. Recently, to meet the urgent requirement of large-scale training set for deep-learning-based co-saliency detection approaches, COCO-SEG has been proposed which are selected from the COCO2017 dataset [39], of which $200,000$ and $8,000$ images are for training and testing respectively from all 78 categories.

We compare our GCAGC method with existing state-of-the-art algorithms in terms of 6 metrics including the

CVPR
#2579

CVPR
#2579

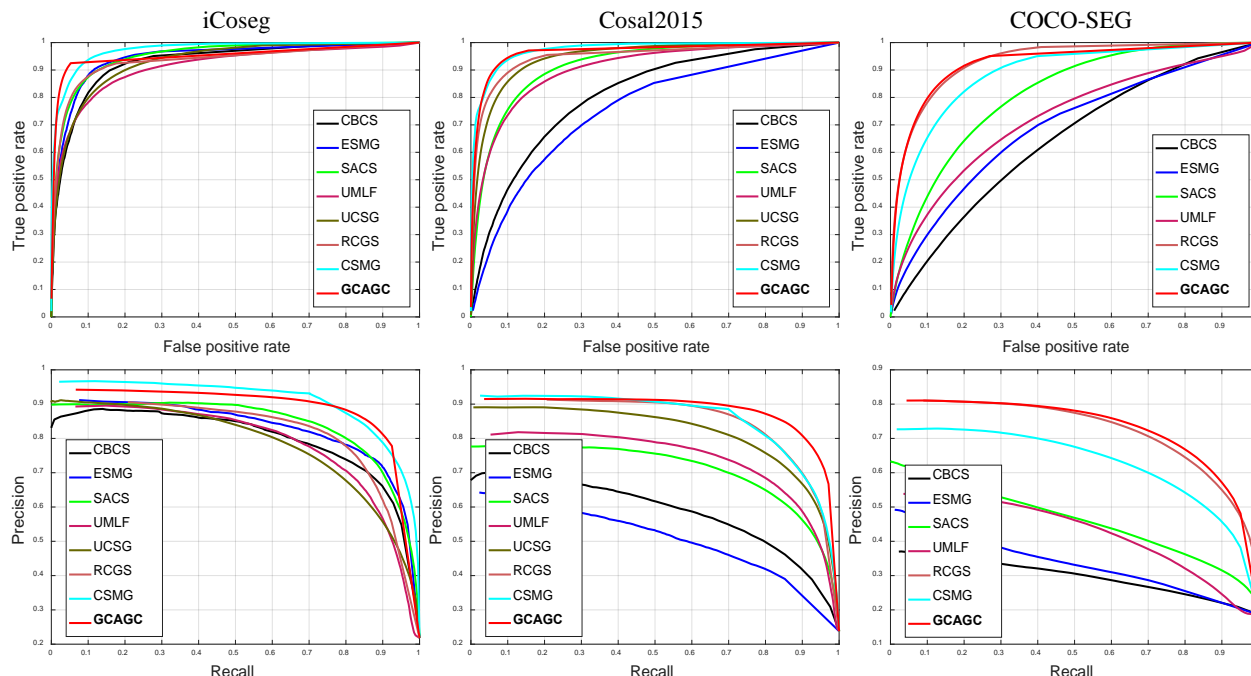CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Figure 6. Comparisons with state-of-the-art methods in terms of PR and ROC curves on three benchmark datasets

Table 1. Statistic comparisons of our GCAGC with the other state-of-the-arts. **Red** and **blue** bold fonts indicate the best and second best performance, respectively.

Methods	iCoseg				Cosal2015				COCO-SEG			
	AP↑	F_β↑	S_m↑	MAE↓	AP↑	F_β↑	S_m↑	MAE↓	AP↑	F_β↑	S_m↑	MAE↓
CBCS [12]	0.7965	0.7408	0.6580	0.1659	0.5859	0.5579	0.5439	0.2329	0.3043	0.3050	0.4710	0.2585
CSHS [42]	0.8454	0.7549	0.7502	0.1774	0.6198	0.6210	0.5909	0.3108	-	-	-	-
ESMG [35]	0.8336	0.7773	0.7677	0.1261	0.5116	0.5120	0.5446	0.2581	0.3387	0.3592	0.4931	0.2349
SACS [5]	0.8399	0.7978	0.7523	0.1516	0.7076	0.6927	0.6938	0.1920	0.4176	0.4234	0.5229	0.3271
CODW [72]	0.8766	0.7990	0.7500	0.1782	0.7437	0.7051	0.6473	0.2733	-	-	-	-
DIM [71]	0.8773	0.7919	0.7583	0.1739	0.6305	0.6162	0.5907	0.3123	0.3043	0.3353	0.4572	0.3871
UMLF [20]	0.7881	0.7148	0.7033	0.2389	0.7444	0.7016	0.6604	0.2687	0.4347	0.4309	0.4872	0.3953
UCSG [24]	**0.9112**	0.8503	0.8200	0.1182	0.8149	0.7589	0.7506	0.1581	-	-	-	-
RCGS [57]	0.8269	0.7730	0.7810	**0.0976**	**0.8573**	0.8097	**0.7959**	**0.0999**	**0.7309**	**0.6814**	**0.7185**	**0.1239**
CSMG [76]	**0.9097**	**0.8517**	**0.8208**	0.1050	0.8569	**0.8216**	0.7738	0.1292	0.6309	0.6208	0.6517	0.1461
GCAGC	0.8867	**0.8532**	**0.8205**	**0.0757**	**0.8799**	**0.8428**	**0.8224**	**0.0890**	**0.7323**	**0.7092**	**0.7294**	**0.1097**

precision-recall (PR) curve [70], the receive operator characteristic (ROC) curve [70], the average precision (AP) score [70], F-measure score F_β [70], S-measure score S_m [11] and Mean Absolute Error (MAE) [57].

4.3. Comparisons with State-of-the-arts

We compare our GCAGC approach with 10 state-of-the-art co-saliency detection methods including CBCS [12], C-SHS [42], ESMG [35], SACS [5], CODW [72], DIM [71], UMLF [20], UCSG [24], RCGS [57], CSMG [76]. For fair comparisons, we directly report available results released by authors or reproduce experimental results by the public

source code for each compared method.

Qualitative Results. Figure 5 shows some visual comparison results with 4 state-of-the-art methods including CBCS [12], ESMG [35], CSMG [76] and RCGS [57]. Our GCAGC can achieve better co-saliency results than the other methods when the co-salient targets suffer from significant appearance variations, strong semantic interference and complex background clutters. In Figure 5, the two left groups of images are selected from iCoseg. Among them, for the group of *Red Sox Players*, the audience in the background share the same semantics with those foreground co-salient players, which makes it very difficult to accurately

CVPR
#2579

CVPR
#2579

CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

Table 2. Ablative studies of our model on iCoseg and Cosal2015. Here GCAGC-N, GCAGC-M, GCAGC-P denote our GCAGC in absence of AGCN, AGCM and the projection matrices **P** in (1), respectively. **Red** bold font indicates the best performance.

Datasets		GCAGC-N	GCAGC-M	GCAGC-P	GCAGC
iCoseg	AP↑	0.8799	0.8606	0.8796	**0.8867**
	F_β↑	0.8504	0.8123	0.8463	**0.8532**
	S_m↑	0.8175	0.8203	0.8122	**0.8205**
	MAE↓	0.0831	0.0796	0.0790	**0.0757**
Cosal2015	AP↑	0.8577	0.8779	0.8737	**0.8799**
	F_β↑	0.8156	0.8373	0.8375	**0.8428**
	S_m↑	0.8167	0.8145	0.8156	**0.8224**
	MAE↓	0.0967	0.0901	0.0851	**0.0890**

differentiate them. Notwithstanding, our GCAGC can accurately highlight the co-salient players due to its two-steps filtering processing from GC filtering to graph clustering that can well preserve spatial consistency while effectively reducing noisy backgrounds. However, the other compared methods cannot achieve satisfying results, which contain either some noisy backgrounds (see the middle columns of RCGS, ESMG, CBCS) or the whole intra-salient areas including non-co-salient regions (see the left-most column of RCGS, the left-fourth columns of ESMG and CBCS). The co-saliency maps in the middle groups (*Apple* and *Monkey*) are generated from the image groups selected from Cosal2015. The *Apple* group suffers from the interferences of other foreground semantic objects such as hand and lemon while the *Monkey* group undergoes complex background clutters. It is obvious that our GCAGC can generate better spatially coherent co-saliency maps than the other methods (see the two bottom rows of ESMG and CBCS, the left-most columns of RCGS and CSMG). The two right-most groups are selected from COCO-SEG, which contain a variety of challenging images with targets suffering from the interferences of various different categories and complicate background clutters. Notwithstanding, our GCAGC can accurately discover the co-salient targets even when they suffer from extremely complicate background clutters (see the *Broccoli* group). The experimental results show that our GCAGC can achieve favorable performance against various challenging factors, validating the effectiveness of our GCAGC model that can adapt well to a variety of complicate scenarios.

Quantitative Results. Figure 6 shows the PR and the ROC curves of all compared methods on three benchmark datasets. We can observe that our GCAGC outperforms the other state-of-the-art methods on three datasets. Especially, all the curves on the largest and most challenging Cosal2015 and COCO-SEG are much higher than the other methods. Meanwhile, Table 1 lists the statistic analysis, among which the RCGS is a representative end-to-end deep-learning-based method that achieves state-of-the-art

performance on both Cosal2015 and COCO-SEG with the F-scores of 0.8097 and 0.6814, respectively. Our GCAGC achieves the best F-scores of 0.8428 and 0.7092 on Cosal2015 and COCO-SEG, respectively, outperforming the second best-performing CSMG by 3.31% on Cosal2015 and RCGS by 2.78% on COCO-SEG. All the qualitative results further demonstrate the effectiveness of jointly learning the GCGAC model that is essential to accurate co-saliency detection.

4.4. Ablative Studies

Here, we conduct ablative studies to validate the effectiveness of the proposed two modules (AGCN and AGCM) and the adaptive graph learning strategy in the AGCN. Table 2 lists the corresponding quantitative statistic results in terms of AP, F_β, S_m and MAE.

First, without AGCN, the GCAGC-N shows obvious performance drop on Cosal2015 in terms of all metrics, especially for both AP and F_β, where the former drops from 0.8799 to 0.8577 by 2.22% and the latter drops from 0.8428 to 0.8156 by 2.72%. Besides, the performance of GCAGC-N on iCoseg also suffers from drop in terms of all metrics.

Second, without AGCM, the GCAGC-M suffers from obvious performance drop in terms of all metrics on both datasets, especially for AP and F_β on iCoseg, where the AP score and the F_β decline from 0.8867 to 0.8606 by 2.61% and from 0.8532 to 0.8123 by 4.09%, respectively. The results validate the effectiveness of the proposed AGCM that can well discriminate the co-objects from all the salient foreground objects to further boost the performance.

Finally, without adaptive graph learning in AGCN, all metrics in GCAGC-P have obvious decline on both datasets, further showing the superiority of proposed AGCN to learn an adaptive graph structure tailored to the co-saliency detection task compared with the fixed graph design in the vanilla GCNs [29].

5. Conclusion

This paper has presented an adaptive graph convolutional network with attention graph clustering for co-saliency detection, mainly including two key designs: an AGCN and an AGCM. The AGCN has been developed to extract long-range dependency cues to characterize the intra- and inter-image correspondence. Meanwhile, to further refine the results of the AGCN, the AGCM has been designed to discriminate the co-objects from all the salient foreground objects in an unsupervised fashion. Finally, a unified framework with encoder-decoder structure has been implemented to jointly optimize the AGCN and the AGCM in an end-to-end manner. Extensive evaluations on three largest and most challenging benchmark datasets including iCoseg, Cosal2015 and COCO-SEG have demonstrated superior performance of the proposed method over the state-of-the-art methods in terms of most metrics.

CVPR
#2579

CVPR
#2579

CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

References

[1] J. Atwood and D. Towsley. Diffusion-convolutional neural networks. In *NeurIPS*, pages 1993–2001, 2016. 2

[2] D. A. Bader, H. Meyerhenke, P. Sanders, and D. Wagner. *Graph partitioning and graph clustering*, volume 588. American Mathematical Soc., 2013. 3

[3] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, pages 3169–3176, 2010. 6

[4] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 2

[5] X. Cao, Z. Tao, B. Zhang, H. Fu, and W. Feng. Self-adaptively weighted co-saliency detection via rank constraint. *TIP*, 23(9):4175–4186, 2014. 2, 7

[6] X. Chen, L.-J. Li, L. Fei-Fei, and A. Gupta. Iterative visual reasoning beyond convolutions. In *CVPR*, pages 7239–7248, 2018. 2

[7] R. Cong, J. Lei, H. Fu, M.-M. Cheng, W. Lin, and Q. Huang. Review of visual saliency detection with comprehensive information. *TCSVT*, 2018. 1

[8] M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, pages 3844–3852, 2016. 2

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[10] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *TPAMI*, 29(11):1944–1957, 2007. 5

[11] D.-P. Fan, M.-M. Cheng, Y. Liu, T. Li, and A. Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, pages 4548–4557, 2017. 7

[12] H. Fu, X. Cao, and Z. Tu. Cluster-based co-saliency detection. *TIP*, 22(10):3766–3778, 2013. 1, 2, 6, 7

[13] H. Fu, D. Xu, S. Lin, and J. Liu. Object-based rgbd image co-segmentation with mutex constraint. In *CVPR*, pages 4428–4436, 2015. 1

[14] H. Fu, D. Xu, B. Zhang, and S. Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, pages 3166–3173, 2014. 1

[15] C. Ge, K. Fu, F. Liu, L. Bai, and J. Yang. Co-saliency detection via inter and intra saliency propagation. *SPIC*, 2016. 2

[16] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *ICML*, pages 1263–1272, 2017. 2

[17] M. Girvan and M. E. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002. 2

[18] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *IJCNN*, volume 2, pages 729–734, 2005. 2

[19] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling. Scene graph generation with external knowledge and image reconstruction. In *CVPR*, pages 1969–1978, 2019. 2

[20] J. Han, G. Cheng, Z. Li, and D. Zhang. A unified metric learning-based framework for co-saliency detection. *TCSVT*, 2017. 7

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[22] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015. 4

[23] K.-J. Hsu, Y.-Y. Lin, and Y.-Y. Chuang. Deepco3: Deep instance co-segmentation by co-peak search and co-saliency detection. In *CVPR*, pages 8846–8855, 2019. 3, 4

[24] 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 *ECCV*, 2018. 1, 2, 3, 4, 7

[25] K. R. Jerripothula, J. Cai, and J. Yuan. Cats: Co-saliency activated tracklet selection for video co-localization. In *ECCV*, pages 187–202, 2016. 1

[26] K. R. Jerripothula, J. Cai, and J. Yuan. Image co-segmentation via saliency co-fusion. *TMM*, 18(9):1896–1909, 2016. 2

[27] B. Jiang, X. Jiang, A. Zhou, J. Tang, and B. Luo. A unified multiple graph learning and convolutional network model for co-saliency estimation. In *MM*, pages 1375–1382, 2019. 3, 4

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[29] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2, 3, 4, 8

[30] L. Landrieu and M. Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, pages 4558–4567, 2018. 2

[31] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *TSP*, 67(1):97–109, 2018. 2

[32] Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, 2018. 3

[33] Q. Li, X.-M. Wu, H. Liu, X. Zhang, and Z. Guan. Label efficient semi-supervised learning via graph filtering. In *CVPR*, pages 9582–9591, 2019. 3, 4

[34] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. In *AAAI*, 2018. 4

[35] Y. Li, K. Fu, Z. Liu, and J. Yang. Efficient saliency-model-guided visual co-saliency detection. *SPL*, 2015. 2, 6, 7

[36] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *ECCV*, pages 335–351, 2018. 2

[37] Z. Li, C. Lang, J. Feng, Y. Li, T. Wang, and S. Feng. Co-saliency detection with graph matching. *TIST*, 10(3):22, 2019. 3, 4

[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

CVPR
#2579

CVPR
#2579

CVPR 2020 Submission #2579. CONFIDENTIAL REVIEW COPY. DO NOT DISTRIBUTE.

[39] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 6

[40] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *TPAMI*, 2011. 6

[41] Z. Liu, W. Zou, L. Li, L. Shen, and O. Le Meur. Co-saliency detection based on hierarchical segmentation. *SPL*, 21(1):88–92, 2013. 1, 2

[42] Z. Liu, W. Zou, L. Li, L. Shen, and O. Le Meur. Co-saliency detection based on hierarchical segmentation. *SPL*, 2014. 7

[43] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[44] M. Narasimhan, S. Lazebnik, and A. Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In *NeurIPS*, pages 2654–2665, 2018. 2

[45] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. In *ICML*, pages 2014–2023, 2016. 2

[46] S. Pan, R. Hu, S.-f. Fung, G. Long, J. Jiang, and C. Zhang. Learning graph embedding with adversarial training methods. *arXiv preprint arXiv:1901.01250*, 2019. 3

[47] A. Papushoy and A. G. Bors. Image retrieval based on query by saliency content. *DSP*, 2015. 1

[48] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *ICCV*, pages 5199–5208, 2017. 2

[49] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *TNN*, 20(1):61–80, 2008. 2

[50] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000. 5

[51] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 6

[52] Y. Sun, J. Han, J. Gao, and Y. Yu. itopicmodel: Information network-integrated topic modeling. In *ICMD*, pages 493–502, 2009. 2

[53] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014. 1

[54] C.-C. Tsai, K.-J. Hsu, Y.-Y. Lin, X. Qian, and Y.-Y. Chuang. Deep co-saliency detection via stacked autoencoder-enabled fusion and self-trained cnns. *TMM*, 2019. 2

[55] 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. *TIP*, 28(1):56–71, 2018. 1, 3, 4

[56] C. Wang, S. Pan, R. Hu, G. Long, J. Jiang, and C. Zhang. Attributed graph clustering: A deep attentional embedding approach. *IJCAI*, 2019. 3

[57] C. Wang, Z.-J. Zha, D. Liu, and H. Xie. Robust deep co-saliency detection with group semantic. 2019. 1, 2, 6, 7

[58] W. Wang, X. Lu, J. Shen, D. J. Crandall, and L. Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, pages 9236–9245, 2019. 2

[59] W. Wang, J. Shen, H. Sun, and L. Shao. Video co-saliency guided co-segmentation. *TCSVT*, 28(8):1727–1736, 2017. 1

[60] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang. Community preserving network embedding. In *AAAI*, 2017. 3

[61] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018. 1, 2, 4

[62] X. Wang and A. Gupta. Videos as space-time region graphs. In *ECCV*, pages 399–417, 2018. 4

[63] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. *TOG*, 38(5):146, 2019. 2

[64] L. Wei, S. Zhao, O. E. F. Bourahla, X. Li, and F. Wu. Group-wise deep co-saliency detection. *arXiv preprint arXiv:1707.07381*, 2017. 1, 2, 6

[65] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019. 2

[66] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 2

[67] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph r-cnn for scene graph generation. In *ECCV*, pages 670–685, 2018. 2

[68] L. Yang, B. Geng, Y. Cai, A. Hanjalic, and X.-S. Hua. Object retrieval using visual query context. *TMM*, 2011. 1

[69] L. Ye, Z. Liu, J. Li, W.-L. Zhao, and L. Shen. Co-saliency detection via co-salient object discovery and recovery. *SPL*, 2015. 1, 2

[70] D. Zhang, H. Fu, J. Han, A. Borji, and X. Li. A review of co-saliency detection algorithms: fundamentals, applications, and challenges. *TIST*, 9(4):38, 2018. 7

[71] D. Zhang, J. Han, J. Han, and L. Shao. Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining. *TNNLS*, 27(6):1163–1176, 2015. 7

[72] D. Zhang, J. Han, C. Li, and J. Wang. Co-saliency detection via looking deep and wide. In *CVPR*, 2015. 6, 7

[73] D. Zhang, J. Han, C. Li, J. Wang, and X. Li. Detection of co-salient objects by looking deep and wide. In *CVPR*, 2015. 1

[74] D. Zhang, J. Han, C. Li, J. Wang, and X. Li. Detection of co-salient objects by looking deep and wide. *IJCV*, 120(2):215–232, 2016. 2

[75] D. Zhang, D. Meng, and J. Han. Co-saliency detection via a self-paced multiple-instance learning framework. *TPAMI*, 39(5):865–878, 2016. 1, 2

[76] K. Zhang, T. Li, B. Liu, and Q. Liu. Co-saliency detection via a mask-guided fully convolutional networks with multi-scale label smoothing. In *CVPR*, pages 3095–3104, 2019. 1, 2, 6, 7

[77] X. Zheng, Z.-J. Zha, and L. Zhuang. A feature-adaptive semi-supervised framework for co-saliency detection. In *MM*, pages 959–966, 2018. 3, 4

[78] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018. 2