DeepACG: Co-Saliency Detection via Semantic-aware Contrast Gromov-Wasserstein Distance

Kaihua Zhang[1], Mingliang Dong[2], Bo Liu[3]*, Xiao-Tong Yuan[1],Qingshan Liu[1]
[1]School of Computer and Software, [2]School of Automation
Nanjing University of Information Science and Technology, Nanjing, China
[3]JD Digits, Mountain View, CA, USA
{zhkhua,kfliubo}@gmail.com

Abstract

The objective of co-saliency detection is to segment the co-occurring salient objects in a group of images. To address this task, we introduce a new deep network architecture via semantic-aware contrast Gromov-Wasserstein distance (DeepACG). We first adopt the Gromov-Wasserstein (GW) distance to build dense 4D correlation volumes for all pairs of image pixels within the image group. These dense correlation volumes enable the network to accurately discover the structured pair-wise pixel similarities among the common salient objects. Second, we develop a semantic-aware co-attention module (SCAM) to enhance the foreground co-saliency through predicted categorical information. Specifically, SCAM recognizes the semantic class of the foreground co-objects, and this information is then modulated to the deep representations to localize the related pixels. Third, we design a contrast edge-enhanced module (EEM) to capture richer contexts and preserve fine-grained spatial information. We validate the effectiveness of our model using three largest and most challenging benchmark datasets (Cosal2015, CoCA, and CoSOD3k). Extensive experiments have demonstrated the substantial practical merit of each module. Compared with the existing works, DeepACG shows significant improvements and achieves state-of-the-art performance.

1. Introduction

Salient object detection mimics the human vision system to identify the most visually distinctive regions in a single image. Extending this task, co-saliency detection (CoSD) is a recently emerging research topic to discover the common salient foreground objects among an image group. Due to its useful potential, it has been increasingly applied into various vision applications, including image/video segmentation [12, 44, 56, 14], object co-localization [41], and weakly supervised semantic segmentation [40].

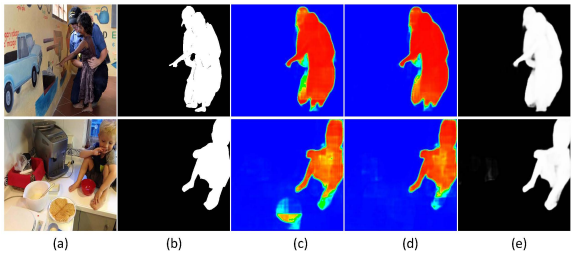

Figure 1. Results of different module variants. (a) Input images; (b) Ground truth; (c) DeepACG w/o SCAM&EEM; (d) DeepACG w/o EEM; (e) The proposed DeepACG.

CoSD has traditionally been formulated as a two-step procedure. First, visual representations are described using hand-engineered features, including: 1) low-level features, such as SIFT [2], color feature [29], and texture feature [21]; 2) mid-level attributes [27]; and 3) multi-cue fusion [1]. Second, these features are then fed into a subsequent module to identify co-saliency. Nonetheless, the hand-engineered features are shallow in nature, and are not able to adapt to large variations of object appearances [58] and complex background textures [43]. Recent studies first improve the CoSD by developing deep-learning-based approaches [57, 46] to extract robust and richer visual representations and explore the semantic correlations between images. These methods have been shown as a promising alternative to conventional approaches. Later, the end-to-end deep learning frameworks [17, 43] have been proposed to integrate the process of feature learning and saliency map prediction. Deep graph neural network has also been adopted to model the non-local and long-range de-

*Corresponding author. This work is supported in part by National Major Project of China for New Generation of AI (No. 2018AAA0100400), in part by the NSFC (61876088, 61876090, 61825601, U20B2065), in part by the 333 High-level Talents Cultivation Project of Jiangsu Province (BRA2020291).

pendencies for CoSD [58]. Although these studies have made a remarkable progress and shown state-of-the-art performance, challenges still exist for further research. The first key question is how to design effective architectures to capture more accurate pixel-pair correspondences while incorporating structured information. Second, the semantic categories of the co-occurring salient objects are usually unknown, but the intra-class differences of shape and appearances are huge. Third, most existing CoSD works focus mainly on the region accuracy, but lose the fine-grained information on boundaries.

Towards addressing the aforementioned challenges, we present a novel deep network architecture via semantic-aware contrast Gromov-Wasserstein distance (DeepACG) for CoSD. Figure 1 illustrates the effectiveness of DeepACG. Gromov-Wasserstein (GW) distance is a notation of distance among metric measure spaces [32, 31, 39]. GW distance is mostly related to the Earth Mover's Distance (EMD) [35] that is widely applied in various classic vision tasks [35, 61, 51]. EMD is constructed between distributions on the same geometric domain, which measures the structural similarity. Differently, GW distance is built between different geometric domains [37]. It is able to measure distances between pairs of nodes within each domain, as well as measuring how these distances compare to those in the counterpart domain [3]. GW distance can extract soft matches in the presence of diverse geometric structures [37]. It has been shown great success in finding correspondences between a source domain and target domain with shared (semantic) structures in both 2D and 3D settings [37]. We adopt GW distance to capture pair-wise correspondence for each pixel feature between the target image and source images in the group (Figure 1(c)). Then, we utilize the semantic categorical information of the co-salient objects to enhance the localization of pixels (Figure 1(d)). In the end, a contrast edge-aware design is used to preserve the boundary information and further improve the segmentation accuracy (Figure 1(e)).

Our major contributions are summarized as follows:

(1) We propose to adopt GW distance to extract dense 4D correlation volumes for all pairs of image pixels and find their correspondences between target and source image domains. With the GW distance, the network is able to minimize distortion of long-and short-range distances, and find the probabilistic matches. The GW distance matching layer can be embedded into the network for end-to-end training.

(2) We present a Semantic-aware Co-Attention Module (SCAM) to enhance the co-occurring salient regions. S-CAM first predicts the semantic categories of the co-salient objects. Then, this information is modulated to the feature representations to refine the localized semantic regions.

(3) We introduce a contrast Edge-Enhanced Module (EEM) to generate fine-grained segmentation for the boundaries of the co-salient objects. To our best knowledge, this is the first edge-aware design in CoSD task.

(4) Extensive experiments have been conducted to validate the effectiveness of our DeepACG on three largest and most challenging datasets, including Cosal2015 [54], CoCA [59], and CoSOD3k [9]. Our DeepACG significantly outperforms the baseline models, and achieves state-of-the-art performance.

2. Related Work

2.1. Image Co-saliency Detection

Early CoSD methods extract image low-level features like Gabor and SIFT features, and then detect image co-saliency through low-level feature consistency between the testing images [2]. The use of mid-level features, such as single image saliency detection result and over-segmentation result can be referred to the literature [21, 27, 1, 16]. With the extracted features, the inter-image saliency is detected by bottom-up or top-down method [52]. The top-down methods generally score image pixels or super-pixels with hand-crafted co-saliency cues [20, 54, 38]. Top-down methods discover the co-saliency from the image feature through proper learning mechanism design. Typical examples include the self-paced multiple-instance learning model [55] and unified metric learning model [13].

More recently, there is a surge of deep learning-based image CoSD models that learns feature extraction and predictor holistically [10]. By treating the testing images as graphical model nodes, single-image saliency detection and cross-image co-occurrence region discovery are formulated into unary and pairwise terms of a fully-connected conditional random field model in [15]. Zhang *et al*. [59] propose a gradient-induced model that utilizes the image gradient information to induce more attention to the discriminative co-salient features. In [57], a hierarchical framework is proposed for CoSD, in which the initial CoSD result generated by neural network model is refined by label smoothing. In [58], a deep graph neural network model is proposed to characterize the intra- and inter-image region correspondence for CoSD.

2.2. Image Matching

Image matching, *i.e.* to establish object or region correspondences between images, is a long-standing research area in computer vision [30]. The image matching technique has extensive applications including SLAM [5], image stitching [25] and structure from motion [6]. Graph matching is one of the major image matching methodologies. Graph matching represents image pixels or key points of one image as graph nodes and the matching task is to estimate the node edge connection between two graphs. One general formulation of the edge estimation task

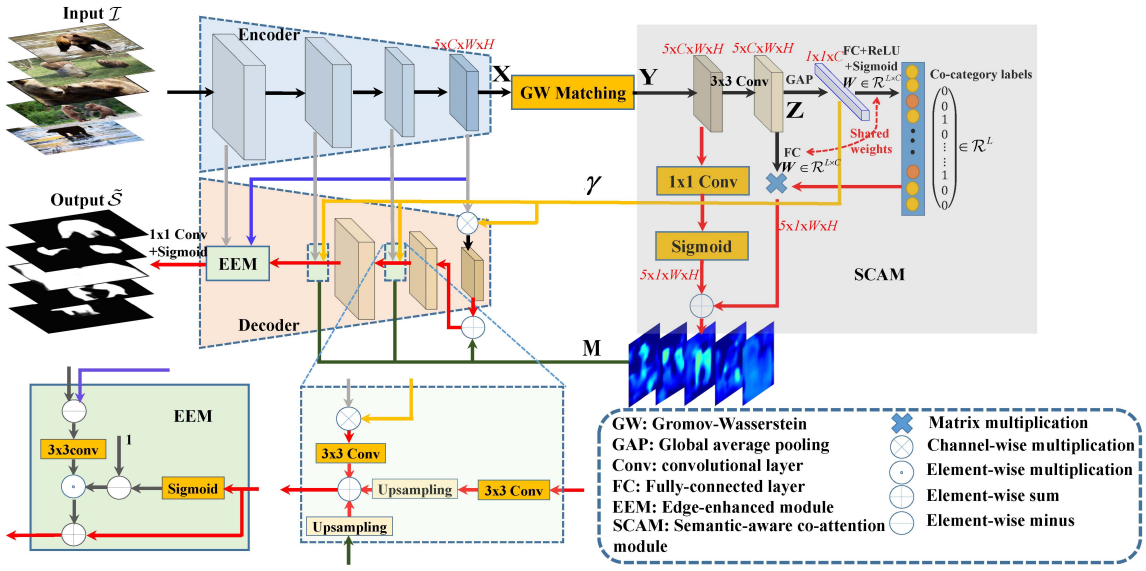

Figure 2. Pipeline of the proposed DeepACG. First, the input images $\mathcal{I}$ are passed through an encoder sub-network, producing the corresponding multi-scale feature presentations. The top-layer features $\mathbf{X}$ are fed into a GW matching layer, which finds the dense correspondence between any pair of local regions in $\mathcal{I}$, and then transfers the matched feature information from the source to the target image domains. Afterwards, the output enhanced features $\mathbf{Y}$ are fed into the SCAM, where we leverage the co-category label information as guidance to learn a semantic co-salient object embedding $\boldsymbol{\gamma}$ and a group of co-attention maps $\mathbf{M}$ that highlight the semantic-aware co-salient regions. $\boldsymbol{\gamma}$ and $\mathbf{M}$ are then fed into a decoder sub-network, which is similar to the feature pyramid network (FPN) [24] that fuses the multi-scale features along the right-to-left path and top-down connections. Finally, the left-layer fused features are passed into the EEM, followed by a 1×1 convolutional layer and a Sigmoid layer to produce the boundary-aware co-saliency maps $\tilde{\mathcal{S}}$.

is the quadratic assignment problem [19, 28], which is NP-hard for an exact solution. A common practice is to solve the task with proper relaxations, such as convex relaxation [4], convex-to-concave relaxation [50] or continuous relaxation [42]. In our work, the proposed GW matching layer is motivated by [37]. In that work, Solomon *et al*. [37] propose a probabilistic matching algorithm through optimizing an entropy-regularized GW objective for shape correspondence.

3. Proposed Approach

Given a group of N relevant images $\mathcal{I} = \{\boldsymbol{I}^n\}_{n=1}^N$ as input, our objective is to learn the DeepACG model that can highlight the common objects with the same category therein. Figure 2 shows the architecture of the DeepACG that is mainly composed of four components: an encoder that leverages the VGG16 network [36] as backbone to extract features; a GW matching layer that aligns the features of the co-salient regions; an SCAM that enhances the foreground co-saliency through predicted categorical information; and a decoder that includes an EEM to produce the boundary-aware co-saliency maps. Our key designs are on the later three components, which will be detailed in the following sections.

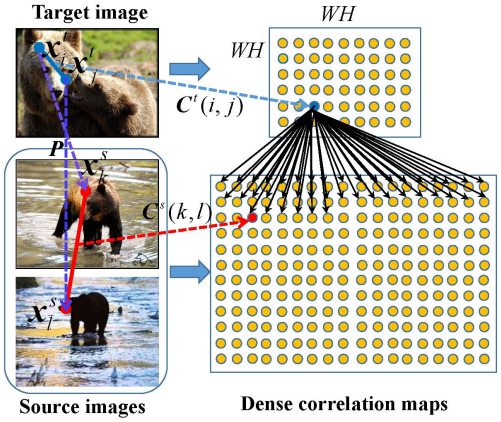

Figure 3. Schematic of GW matching. For a feature vector in the target image, we take the inner product with all pairs in itself and generate a 4D $W \times H \times W \times H$ correlation volume, where each pixel produces a 2D response map. Here we reshape the correlation volume to a $WH \times WH$ correlation map $\boldsymbol{C}^t$. We apply similar strategy to the source images and produce an $NWH \times NWH$ ($N = 2$ here) correlation map $\boldsymbol{C}^s$. Finally, we construct the GW distance using the two correlation maps for structural matching.

3.1. GW Matching Layer

Each image in $\mathcal{I}$ is fed into the encoder sub-network, producing its corresponding feature representation $\boldsymbol{X} =$

$[\boldsymbol{x}_1^\top; \ldots; \boldsymbol{x}_{WH}^\top] \in \mathcal{R}^{WH \times C}$, where W, H denote the width and height of the feature map, C is the channel number, and $\boldsymbol{x}_i \in \mathcal{R}^C$ denotes the i-th feature vector. We sequentially select one image as the target image, and the other $N-1$ images in $\mathcal{I}$ as the source images. We then use their features to conduct GW matching between the target image and the source ones. Specifically, as shown in Figure 3, given the target image features $\boldsymbol{X}^t = [\boldsymbol{x}_1^t, \ldots, \boldsymbol{x}_{WH}^t]^\top \in \mathcal{R}^{WH \times C}$ and the source image representations $\boldsymbol{X}^s = [\boldsymbol{x}_1^s, \ldots, \boldsymbol{x}_{(N-1)WH}^s]^\top \in \mathcal{R}^{(N-1)WH \times C}$, we first compute their corresponding correlation maps as $\boldsymbol{C}^t = \boldsymbol{X}^t \boldsymbol{X}^{t\top}$, $\boldsymbol{C}^s = \boldsymbol{X}^s \boldsymbol{X}^{s\top}$, where the (i,j)-th element of $\boldsymbol{C}^t$ and the (k,l)-th element of $\boldsymbol{C}^s$ are formulated as

$$\boldsymbol{C}^t(i,j) = \boldsymbol{x}_i^{t\top} \boldsymbol{x}_j^t, \boldsymbol{C}^s(k,l) = \boldsymbol{x}_k^{s\top} \boldsymbol{x}_l^s, \tag{1}$$

which are the inner product of feature pairs at locations i, j on the target domain and at locations k, l on the source domain, measuring the dependency between any location pairs.

If there exist matching pairs between the target and the source domains, i.e. $i \mapsto k$ and $j \mapsto l$, the distance between the feature pair at locations i and j on the target domain should be similar to that at locations k and l on the source domain. Based on this assumption, we leverage the regularized 2-GW distance [37] for optimal structural matching, which is defined as

$$\begin{aligned} &GW_2^2(\boldsymbol{C}^s, \boldsymbol{C}^t) \\ &= \min_{\boldsymbol{P} \in \mathcal{P}} \{ \sum_{ijkl} (\boldsymbol{C}^t(i,j) - \boldsymbol{C}^s(k,l))^2 \boldsymbol{P}(i,k) \boldsymbol{P}(j,l) - \alpha H(\boldsymbol{P}) \}, \end{aligned} \tag{2}$$

where $H(\boldsymbol{P}) = -\sum_{ik} \boldsymbol{P}(i,k) \ln(\boldsymbol{P}(i,k))$ is the entropy of the optimal matching flows $\boldsymbol{P} \in \mathcal{R}_+^{WH \times (N-1)WH}$, set $\mathcal{P} = \{\boldsymbol{P} : \boldsymbol{P}\boldsymbol{1}_{(N-1)WH} = \boldsymbol{1}_{WH}, \boldsymbol{P}^\top \boldsymbol{1}_{WH} = \boldsymbol{1}_{(N-1)WH}\}$, $\boldsymbol{1}_D$ is a D-dimensional all-ones vector. Intuitively, the matching flow $\boldsymbol{P}(i,k)$ represents the probability that the i-th location in the target domain corresponds to the k-th location in the source domain. After achieving the optimal matching flows $\boldsymbol{P}$, we transfer the aligned feature information from the source domains to the target domain via

$$\widetilde{\boldsymbol{X}}^t = \boldsymbol{P}\boldsymbol{X}^s. \tag{3}$$

Finally, we concatenate and reshape all the aligned features $\widetilde{\boldsymbol{X}}_i^t, i = 1, \ldots, N$, producing the strong features $\mathbf{Y} = \mathrm{cat}(\widetilde{\boldsymbol{X}}_1^t, \ldots, \widetilde{\boldsymbol{X}}_N^t) \in \mathcal{R}^{N \times C \times W \times H}$ that are effective to enhance the co-saliency regions in $\mathcal{I}$.

As listed by Algorithm 1, we directly use the GW solver proposed by [37] to solve problem (2), which alternates between a closed-form exponential formula and Sinkhorn projection (refer to Algorithm 2) onto the cone of doubly stochastic matrices. The GW mathcing layer is differentiable since its operations in Algorithms 1 and 2 only contain matrix-vector multiplication and element-wise division, which can be readily plugged into the vanilla deep neural networks for end-to-end training. The GW matching layer can be readily implemented with automatic differentiation in PyTorch [33].

Algorithm 1 GW solver

Input: $\boldsymbol{C}^t, \boldsymbol{C}^s, \alpha, \eta = 0.5$
Output: $\boldsymbol{P}$

1: $\boldsymbol{P} \leftarrow \mathrm{Ones}(WH, (N-1)WH)$
2: **for** $i = 1, 2, 3, \ldots$ **do**
3: $\quad \boldsymbol{K} \leftarrow \exp\left(\boldsymbol{C}^t \boldsymbol{P} \boldsymbol{C}^{s\top} / \alpha\right)$
4: $\quad \boldsymbol{P} \leftarrow$ Sinkhorn-projection$(\boldsymbol{K}^{\wedge\eta} \odot \boldsymbol{P}^{\wedge(1-\eta)})$, where $\wedge\eta$ denotes the element-wise power of η of a matrix, $\odot$ denotes element-wise multiplication.
5: **end for**

Algorithm 2 Sinkhorn-projection

Input: $\boldsymbol{K}$
Output: $\mathrm{diag}(\boldsymbol{v})\boldsymbol{K}\mathrm{diag}(\boldsymbol{w})$

1: $\boldsymbol{v}, \boldsymbol{w} \leftarrow \boldsymbol{1}$
2: **for** $j = 1, 2, 3, \ldots$ **do**
3: $\quad \boldsymbol{v} \leftarrow \boldsymbol{1} \oslash (\boldsymbol{K}\boldsymbol{w})$, where $\oslash$ denotes element-wise division
4: $\quad \boldsymbol{w} \leftarrow \boldsymbol{1} \oslash (\boldsymbol{K}^\top \boldsymbol{v})$
5: **end for**

Figure 4. Illustrations of the behavior of the learned matching flows $\boldsymbol{P}$. The starting points of arrows represent the anchor locations in the target image, and the ending points represent their corresponding matched locations in the source images.

Figure 4 visualizes two examples of the behavior of the learned matching flows $\boldsymbol{P}$, where it has found the meaningful relational cues across the source images.

3.2. Semantic-aware Co-attention Module

The GW matching is effective to enhance the co-saliency regions via structurally learning the dense correspondence between all feature pairs in $\mathcal{I}$. However, in some challenging scenarios, where there exist distractors with similar appearance to the co-salient targets (see the *bananas* vs. the *peanut butter jar* in Figure 5), the misleading matching flows between the distractors and the co-salient targets may cause the aligned features to highlight the distractors (see the left-second column in Figure 5, where the distractor *peanut butter jar* is highlighted). To address this issue, we further propose the SCAM to guide the features to tell

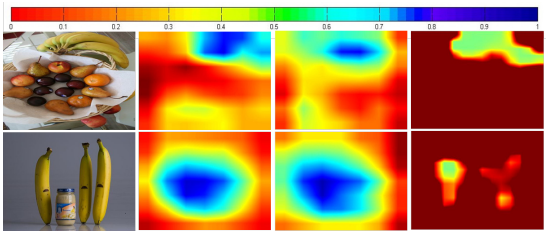

Figure 5. Effect of SCAM to distractor (i.e. *peanut butter jar*). Left to right: inputs, the responses of features $\mathbf{Y}$, the correlation responses of the embedding $\boldsymbol{\gamma}$ and the features $\mathbf{Y}$, the co-attention responses $\mathbf{M}$.

the co-salient targets from distractors with predicted semantic co-category information.

As shown by Figure 2, given the input $\mathbf{Y} \in \mathcal{R}^{N\times C\times W\times H}$ of the SCAM, we first feed it into a 3×3 convolutional layer, producing the features $\mathbf{Z} = \mathrm{Conv}_{3\times3}(\mathbf{Y})$, following a GAP layer to produce the semantic co-salient object embedding $\boldsymbol{\gamma} \in \mathcal{R}^{C}$

$$\boldsymbol{\gamma} = \frac{1}{NWH}\sum_{n=1}^{N}\sum_{w=1}^{W}\sum_{h=1}^{H}\mathbf{Z}(n,:,w,h). \tag{4}$$

The embedding $\boldsymbol{\gamma}$ is then passed through an FC layer with weights $\boldsymbol{W} \in \mathcal{R}^{L\times C}$, where L denotes the number of categories, following a ReLU and a Sigmoid layers, yielding the predicted co-category labels $\tilde{\boldsymbol{l}} = \mathrm{Sigmoid}(\mathrm{ReLU}(\boldsymbol{W}\boldsymbol{\gamma})) \in \mathcal{R}^{L}$. Then, we take the shared FC weights $\boldsymbol{W}$ as a linear classifier to classify the features $\mathbf{Z}$, yielding the classification results $\tilde{\boldsymbol{M}} \in \mathcal{R}^{NWH\times L}$. Finally, we use the predicted co-category labels $\tilde{\boldsymbol{l}}$ to fuse the classification results, producing the semantic-guided co-attention response $\mathbf{M} = \mathrm{reshape}(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{l}}) \in \mathcal{R}^{N\times W\times H}$.

Since the classifier pays more attention to the most discriminative features for semantic classification, the learned co-attention response $\mathbf{M}$ may omit some useful information that is essential to highlight the co-salient regions. To solve this problem, we design a residual module that complementarily learns an another spatial attention map from the input features $\mathbf{X}$ using a 1×1 convolutional layer and a Sigmoid layer

$$\mathbf{M} = \mathbf{M} \oplus \mathrm{Sigmoid}(\mathrm{Conv}_{1\times1}(\mathbf{Y})), \tag{5}$$

where $\oplus$ denotes element-wise sum operator.

Figure 5 shows the effect of $\boldsymbol{\gamma}$ and $\mathbf{M}$, where we can observe that $\boldsymbol{\gamma}$ can weaken the influence of the distractor while enhancing the co-saliency features, and $\mathbf{M}$ can further filter out the distractors. Finally, the learned $\boldsymbol{\gamma}$ in (4) and $\mathbf{M}$ in (5) are fed into the decoder sub-network, which are used to modulate the multi-scale features, such that $\boldsymbol{\gamma}$ serves as the channel-wise scale parameters adjusting the weights of different channels in the feature maps, and $\mathbf{M}$ is used as the element-wise bias parameters injecting spatial co-attention prior to the modulated features [48].

3.3. Decoder Sub-network

The decoder sub-network has the similar architecture as the FPN [24], which combines low-resolution, semantically strong features with high-resolution, semantically weak features via a right-to-left pathway and top-down connections to the corresponding encoder layers. Besides, the embedding $\boldsymbol{\gamma}$ in (4) and the co-attention maps $\mathbf{M}$ in (5) are fed into the decoder sub-network to modulate each layer of features. Finally, the left-layer features are passed through the edge-enhanced module for co-salient object boundary enhancement, following a 1×1 convolutional layer and a Sigmoid layer, producing the predicted co-saliency maps $\tilde{\mathcal{S}} = \{\tilde{\boldsymbol{S}}^{n}\}_{n=1}^{N}$.

Edge-Enhanced Module (EEM): Due to the downsampling of the input images, their high-level semantic features pay more attention to the inside parts of the objects rather than their boundaries. Especially after the features are further spatially modulated using the co-attention maps $\mathbf{M}$, the object boundary information loses considerably, leading to inaccurately predicted results, especially on the object boundaries. To address this issue, we further design the EEM for boundary enhancement, which can effectively fuse the rich contexts from high-level features and the fine-grained spatial details from the low-level ones. We first resize the high-level feature maps to have the same size as the low-level ones, and then we calculate the difference between the two feature maps, producing the boundary features with rich contexts and spatial detail information. To further enhance the boundary features, we leverage a residual module that learns the residual to weight the boundary features. Finally, the enhanced boundary features and the input features from the right path are fused to generate the output of the EEM.

Loss: The whole network parameters are optimized end-to-end using the loss function

$$\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \mathcal{L}_{\mathrm{IoU}} + \mathcal{L}_{\mathrm{WBCE}}, \tag{6}$$

where $\mathcal{L}_{\mathrm{cls}}$ is the cross-entropy loss for semantic classification defined as

$$\mathcal{L}_{\mathrm{cls}} = -\frac{1}{L}\sum_{l=1}^{L}\boldsymbol{l}(l)\log(\tilde{\boldsymbol{l}}(l)) - (1-\boldsymbol{l}(l))\log(1-\tilde{\boldsymbol{l}}(l)), \tag{7}$$

where $\boldsymbol{l} \in \mathcal{R}^{L}$ denotes the ground-truth co-category labels. $\mathcal{L}_{\mathrm{IoU}}$ is the IoU loss that is a relaxation of the Jaccard distance widely used to evaluate segmentation accuracy [23]

$$\mathcal{L}_{\mathrm{IoU}} = 1 - \frac{\sum_{i,j}\min(\boldsymbol{S}(i,j),\tilde{\boldsymbol{S}}(i,j))}{\sum_{i,j}\max(\boldsymbol{S}(i,j),\tilde{\boldsymbol{S}}(i,j))}, \tag{8}$$

where $\boldsymbol{S} \in \{0,1\}^{W\times H}$ is the ground-truth mask of the co-saliency map. $\mathcal{L}_{\mathrm{WBCE}}$ is the Weighted Binary Cross-Entropy (WBCE) loss for pixel-wise classification that is

defined as

$$\mathcal{L}_{\mathrm{WBCE}} = -\frac{1}{WH}\sum_{j=1}^{H}\sum_{i=1}^{W} w\boldsymbol{S}(i,j)\log(\tilde{\boldsymbol{S}}(i,j)) - (1-w)(1-\boldsymbol{S}(i,j))\log(1-\tilde{\boldsymbol{S}}(i,j)), \tag{9}$$

where weight w is the ratio of the negative pixel number over all the pixels in one image, which balances the importance of the positive and negative pixels in the loss.

4. Experiments

4.1. Implementation Details

We use the similar settings as the recently proposed CoSD framework [43] to configure the system: the input image group $\mathcal{I}$ contains $N = 5$ images with the same category as a batch, and a mini-batch of $6 \times \mathcal{I}$ images from all categories are sent into the network simultaneously. All the images are resized to $224 \times 224 \times 3$ pixels as input, and the predicted co-saliency maps are resized to the expected sizes as outputs. In the training process, we use the Adam algorithm [18] to optimize the whole network end-to-end and set the first and second decay rates of momentum to 0.9 and 0.999, respectively. We set the weight attenuation to 1e-6. We set the learning rate of all parameters of the network to 1e-4, and set the learning rate to be an half every $25,000$ iterations until convergence. Our DeepACG is implemented in PyTorch [33] and an NVIDIA RTX2080Ti GPU is adopted for acceleration, which requires a total of $140,000$ training steps.

We use the COCO-SEG dataset released by [43] for training, which contains $200,000$ images and we remove the images containing small objects therein. The dataset includes $L = 78$ categories and each image has a manually-labeled binary mask with co-category labels $\boldsymbol{l}$. The training process takes about 30 hours.

4.2. Datasets and Evaluation Metrics

The deepACG model is evaluated on three largest and most challenging benchmark datasets, including Cosal2015 [54], CoCA [59], and CoSOD3k [9]. Among them, the Cosal2015 is a widely-used benchmark dataset for CoSD. It owns $2,015$ images of 50 categories. For some categories, such as pineapple, there are many non-co-salient objects with similar appearances, which is very challenging to accurately detect the co-salient targets. The CoCA contains 80 classes with $1,297$ images in total. This dataset is characterized by more complex background interferences than those in Cosal2015. The CoSOD3k is the largest evaluation benchmark at present. It has a total of 160 categories with $3,316$ images. Different from Cosal2015, a large amount of images in CoSOD3k have two or three instances to be highlighted, which span a wide range of categories, shapes, object sizes, and backgrounds. We use four evaluation metrics for comparison, including the mean absolute error MAE [43], F-measure F_β [52], E-measure E_m [8], and S-measure S_m [7].

4.3. Comparisons with State-of-the-arts

We leverage the evaluation codes released by Fan *et al.* [9] to compare with several state-of-the-art methods, including BASNet [34], PoolNet [26], EGNet [60], CBCD [11], ESMG [22], CODR [49], DIM [53], C-SMG [57], SSNM [56], GICD [59], GW [46], SCRN [47], and GCAGC [58]. Among them, BASNet [34], PoolNet [26], and EGNet [60] are state-of-the-art saliency object detection methods that have achieved favorable performance on CoSD task.

Qualitative Results. Figure 6 shows some visualization results of our DeepACG compared with three representative state-of-the-art methods, including GICD [59], GCAGC [58], and CSMG [57]. The proposed DeepACG performs favorably well under the challenging scenarios that the co-salient targets suffer from complex background clutters, small sizes, large-scale appearance or shape variations, severe occlusions, etc. In the *Alarm Clock* and *Beaker* groups, the co-salient objects suffer from large-scale shape and appearance variations (see the alarm clock in the rightmost column in *Alarm Clock* and the two beakers in the third and fourth columns in *Beaker*), making it difficult to accurately extract the co-salient targets without semantic guidance. Our DeepACG achieves much better visual results than the others due to the use of predicted semantic information as guidance in the SCAM. In the *Globe* group, the co-salient globes undergo significant appearance variations (the second column) and background clutters (the right-two columns), making only using appearance information unable to well group the co-salient targets (see the results of GCAGC and CSMG). The DeepACG makes use of the correlation maps to conduct GW matching, which naturally encodes the shape topology information in the correlation maps to help better group the co-salient targets with different appearance textures yet similar shapes. As shown by the left-two columns, the co-saliency maps generated by GICD contain a large amount of background noises. The reason is that it uses consensus embedding as guidance for learning, which is polluted by background distractors. Similar results produced by GICD on *Alarm Clock* and *Beaker* groups. Our DeepACG can effectively filter out the distractors by using the co-attention mask in the SCAM. In the *Frog* and the *Pinapple* groups, some parts of the co-salient targets have tenuous and complex boundaries (see the frog feet and the pinapple crown). Due to the use of EEM to enhance boundary information, our DeepACG can produce satisfying co-saliency maps with fine boundary details while the co-saliency maps generated by the compared methods are more coarse.

Figure 6. Results of our DeepACG compared with other state-of-the-art methods

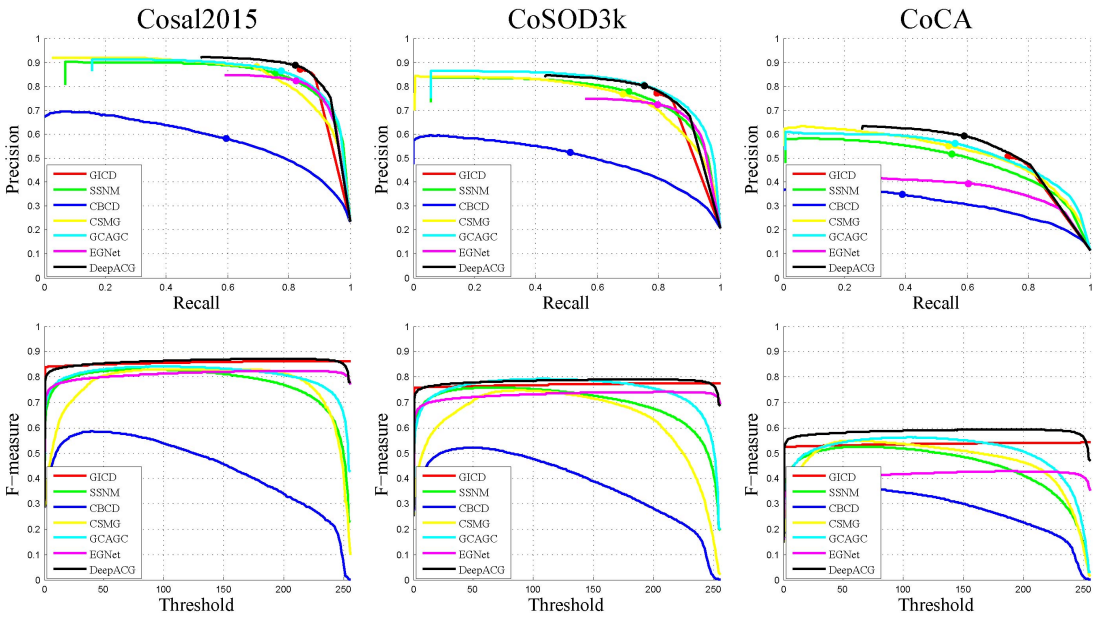

Figure 7. Comparisons with state-of-the-art methods in terms of PR and F-measure curves on three benchmark datasets

Qualitative Results. Figure 7 shows the PR and the F-measure curves of all compared methods on three benchmark datasets. It can be observed that DeepACG achieves the best performance over the other state-of-the-arts, where all the curves of DeepACG are on the top of those generated by the other comparison methods. Meanwhile, Table 1 lists the statistic comparison results of all compared methods, where GICD and GCAGC are the most recently proposed state-of-the-art methods. DeepACG achieves the best performance in terms of all evaluation metrics on three benchmark datasets. Specifically, on the COCA dataset, DeepACG reaches the best scores of 0.552, 0.771, and 0.688 in terms of F-measure, E-measure, and S-measure, respectively, with a gain of 2.9%, 1.7%, 1.7% compared to the second best-performing GCAGC with scores of 0.523, 0.754, and 0.669, respectively. Besides, on the other two benchmarks, DeepACG also achieves the best scores in terms of all metrics, further demonstrating its effectiveness on large-scale

Table 1. Statistic comparisons of our DeepACG with the other state-of-the-arts. **Red** and **blue** bold fonts indicate the best and second-best performance, respectively. *single image saliency object detection methods.-the results have not been provided by the benchmarks.

Methods	Cosal2015				CoSOD3k				CoCA			
	MAE↓	S_m↑	E_m↑	F_β↑	MAE↓	S_m↑	E_m↑	F_β↑	MAE↓	S_m↑	E_m↑	F_β↑
BASNet* (*CVPR2019*) [34]	0.097	0.820	0.846	0.784	0.122	0.753	0.791	0.696	0.195	0.589	0.623	0.397
EGNet* (*ICCV2019*) [60]	0.099	0.818	0.842	0.782	0.119	0.762	0.796	0.703	0.179	0.594	0.637	0.389
PoolNet* (*CVPR2020*) [26]	0.094	0.820	0.851	0.785	0.120	0.763	0.797	0.704	0.179	0.599	0.631	0.401
SCRN* (*CVPR2020*) [47]	0.097	0.814	0.854	0.789	0.118	0.773	0.806	0.717	0.166	0.610	0.658	0.416
CBCD (*TIP2013*) [11]	0.233	0.544	0.656	0.503	0.228	0.528	0.589	0.363	0.172	0.526	0.659	0.313
ESMG (*SPL2014*) [22]	0.247	0.552	0.653	0.470	0.239	0.532	0.615	0.364	-	-	-	-
CODR (*SPL2015*) [49]	0.204	0.689	0.723	0.608	0.229	0.630	0.645	0.458	-	-	-	-
DIM (*TNNLS2016*) [53]	0.312	0.593	0.697	0.559	0.327	0.559	0.610	0.420	-	-	-	-
GW (*IJCAI2017*) [46]	0.147	0.743	0.793	0.697	-	-	-	-	0.171	0.603	0.666	0.398
CSMG (*CVPR2019*) [57]	0.130	0.774	0.818	0.777	0.157	0.711	0.723	0.645	0.124	0.632	0.734	0.503
SSNM (*AAAI2020*) [56]	0.102	0.788	0.843	0.794	0.120	0.726	0.756	0.675	0.116	0.628	0.741	0.482
GCAGC (*CVPR2020*) [58]	0.085	0.817	0.866	0.813	0.100	**0.785**	0.816	0.740	**0.111**	**0.669**	**0.754**	**0.523**
GICD (*ECCV2020*) [59]	**0.071**	**0.842**	**0.884**	**0.834**	**0.089**	0.778	**0.831**	**0.743**	0.125	0.658	0.701	0.504
DeepACG	**0.064**	**0.854**	**0.892**	**0.842**	**0.089**	**0.792**	**0.838**	**0.756**	**0.102**	**0.688**	**0.771**	**0.552**

Table 2. Ablations of our model on the CoCA. NLA is short for non-local attention. EMD is short for earth mover's distance. **Red** bold fonts indicate the best performance.

Models	MAE↓	S_m↑	E_m↑	F_β↑
w/o GW	0.107	0.676	0.756	0.529
w/o SCAM	0.104	0.678	0.764	0.529
w/o GW&SCAM	0.130	0.632	0.715	0.443
w/o EEM	0.105	0.679	0.767	0.532
with NLA	0.105	0.676	0.756	0.532
with EMD	0.104	0.678	0.760	0.535
DeepACG	**0.102**	**0.688**	**0.771**	**0.552**

challenging datasets.

4.4. Ablation Study

To verify the effect of the key module designs in our DeepACG, we further conduct extensive ablative studies on CoCA dataset. Table 2 lists the corresponding experimental results in terms of all metrics. We can observe that without GW matching, the F_β score drops from 0.552 to 0.529 by 2.3% and the E_m score reduces by 1.5% from 0.771 to 0.756, validating that the GW matching plays a vital role in our DeepACG. Moreover, without SCAM, the performance of DeepACG drops significantly in terms of all metrics, especially for the F_β that decreases from 0.552 to 0.529 by 2.3%. If we further remove both the modules GW and SCAM, the performance of our model significantly drops by 10.9% and 5.6% in terms of F-measure and S-measure, respectively. Then, we test our DeepACG without EEM, which suffers from a drop score of 2% in terms of F-measure, which proves the effectiveness of the EEM to handle complex object boundaries that is essential to producing high-quality co-saliency maps.

Finally, we replace the GW matching layer with the NLA module [45], which enhances features by considering all pair-wise location interactions. The DeepACG with NLA has an F_β score of 0.532, which is lower than DeepACG by 2%. This is due to the NLA module generates non-zero weights for all pair-wise positions, which may introduce noisy interactions that degrade the model. However, the deepACG learns the optimal matching flows based on the GW distance, which only assign non-zero weights to the most stable locations (see Figure 4), thereby achieving better performance. So does the DeepACG with EMD [51], which performs better than that with NLA, but worse than DeepACG by a drop of 1.7% in terms of F_β score. This is because the EMD matching directly learns structure similarity between two semantic feature points, which is less robust to the object topology changes than the GW matching.

5. Conclusion

This paper have presented a new deep network architecture DeepACG for co-saliency detection, which includes three novel module designs. First, a novel GW distance matching layer has been designed that is built on dense 4D correlation volumes for all pairs of image pixels within the image group, which is able to accurately discover the structured pair-wise pixel similarities among the co-salient objects. Second, a semantic-aware co-attention module has been developed to enhance the foreground co-saliency through predicted categorical information. Third, a contrast edge-enhanced module has been designed to capture richer context and preserve fine-grained spatial information. Extensive evaluations on three benchmarks have demonstrated superior performance of our method over state-of-the-arts.

References

[1] Xiaochun Cao, Zhiqiang Tao, Bao Zhang, Huazhu Fu, and Wei Feng. Self-adaptively weighted co-saliency detection via rank constraint. *TIP*, 2014. 1, 2

[2] Kai Yueh Chang, Tyng Luh Liu, and Shang Hong Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR*, 2011. 1, 2

[3] Liqun Chen, Zhe Gan, Yu Cheng, Linjie Li, Lawrence Carin, and Jingjing Liu. Graph optimal transport for cross-domain alignment. *arXiv preprint arXiv:2006.14744*, 2020. 2

[4] Qifeng Chen and Vladlen Koltun. Robust nonrigid registration by convex optimization. In *ICCV*, 2015. 3

[5] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *ICRA*, 2012. 2

[6] Bin Fan, Qingqun Kong, Xinchao Wang, Zhiheng Wang, Shiming Xiang, Chunhong Pan, and Pascal Fua. A performance evaluation of local features for image-based 3d reconstruction. *TIP*, 2019. 2

[7] Deng Ping Fan, Ming Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps. In *ICCV*, 2017. 6

[8] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. *arXiv preprint arXiv:1805.10421*, 2018. 6

[9] Deng-Ping Fan, Tengpeng Li, Zheng Lin, Ge-Peng Ji, Dingwen Zhang, Ming-Ming Cheng, Huazhu Fu, and Jianbing Shen. Re-thinking co-salient object detection. *arXiv preprint arXiv:2007.03380*, 2020. 2, 6

[10] Deng-Ping Fan, Zheng Lin, Ge-Peng Ji, Dingwen Zhang, Huazhu Fu, and Ming-Ming Cheng. Taking a deeper look at the co-salient object detection. In *CVPR*, 2020. 2

[11] Huazhu Fu, Xiaochun Cao, and Zhuowen Tu. Cluster-based co-saliency detection. *TIP*, 2013. 6, 8

[12] Huazhu Fu, Dong Xu, Bao Zhang, and Stephen Lin. Object-based multiple foreground video co-segmentation. In *CVPR*, 2014. 1

[13] Junwei Han, Gong Cheng, Zhenpeng Li, and Dingwen Zhang. A unified metric learning-based framework for co-saliency detection. *TCSVT*, 2017. 2

[14] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Deepco3: Deep instance co-segmentation by co-peak search and co-saliency detection. In *CVPR*, 2019. 1

[15] Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, Xiaoning Qian, and Yung-Yu Chuang. Unsupervised cnn-based co-saliency detection with graphical optimization. In *ECCV*, 2018. 2

[16] Bo Jiang, Xingyue Jiang, Jin Tang, and Bin Luo. Co-saliency detection via a general optimization model and adaptive graph learning. *TMM*, 2020. 2

[17] Bo Jiang, Xingyue Jiang, Ajian Zhou, Jin Tang, and Bin Luo. A unified multiple graph learning and convolutional network model for co-saliency estimation. In *MM*, 2019. 1

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Eugene L Lawler. The quadratic assignment problem. *Management science*, 1963. 3

[20] Hongliang Li, Fanman Meng, and King Ngi Ngan. Co-salient object detection from multiple images. *TMM*, 2013. 2

[21] Hongliang Li and King Ngi Ngan. A co-saliency model of image pairs. *TIP*, 2011. 1, 2

[22] Yijun Li, Keren Fu, Zhi Liu, and Jie Yang. Efficient saliency-model-guided visual co-saliency detection. *SPL*, 2014. 6, 8

[23] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018. 5

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*. 3, 5

[25] Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. Smoothly varying affine stitching. In *CVPR*, 2011. 2

[26] Jiang Jiang Liu, Qibin Hou, Ming Ming Cheng, Jiashi Feng, and Jianmin Jiang. A simple pooling-based design for real-time salient object detection. In *CVPR*, 2020. 6, 8

[27] Zhi Liu, Wenbin Zou, Lina Li, Liquan Shen, and Olivier Le Meur. Co-saliency detection based on hierarchical segmentation. *SPL*, 2013. 1, 2

[28] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. A survey for the quadratic assignment problem. *EJOR*, 2007. 3

[29] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 1

[30] Jiayi Ma, Xingyu Jiang, Aoxiang Fan, Junjun Jiang, and Junchi Yan. Image matching from handcrafted to deep features: A survey. *IJCV*, 2020. 2

[31] Facundo Mémoli. The gromov–wasserstein distance: A brief overview. *Axioms*, 2014. 2

[32] Quentin Mérigot. A multiscale approach to optimal transport. *CGF*, 2011. 2

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 4, 6

[34] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *CVPR*, 2019. 6, 8

[35] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *IJCV*, 2000. 2

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[37] Justin Solomon, Gabriel Peyré, Vladimir G Kim, and Suvrit Sra. Entropic metric alignment for correspondence problems. *TOG*. 2, 3, 4

[38] Hangke Song, Zhi Liu, Yufeng Xie, Lishan Wu, and Mengke Huang. Rgbd co-saliency detection via bagging-based clustering. *SPL*, 2016. 2

[39] Karl Theodor Sturm. The space of spaces: curvature bounds and gradient flows on the space of metric measure spaces. *Mathematics*, 2012. 2

[40] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. *arXiv preprint arXiv:2007.01947*, 2020. 1

[41] Kevin Tang, Armand Joulin, Li Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014. 1

[42] Yu Tian, Junchi Yan, Hequan Zhang, Ya Zhang, Xiaokang Yang, and Hongyuan Zha. On the convergence of graph matching: Graduated assignment revisited. In *ECCV*, 2012. 3

[43] Chong Wang, Zheng-Jun Zha, Dong Liu, and Hongtao Xie. Robust deep co-saliency detection with group semantic. In *AAAI*, 2019. 1, 6

[44] Wenguan Wang, Jianbing Shen, Hanqiu Sun, and Ling Shao. Video co-saliency guided co-segmentation. *TCSVT*, 2017. 1

[45] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 8

[46] Lina Wei, Shanshan Zhao, Omar El Farouk Bourahla, Xi Li, and Fei Wu. Group-wise deep co-saliency detection. *IJCAI*, 2017. 1, 6, 8

[47] Zhe Wu, Li Su, and Qingming Huang. Stacked cross refinement network for edge-aware salient object detection. In *ICCV*, 2020. 6, 8

[48] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 5

[49] Linwei Ye, Zhi Liu, Junhao Li, Wan Lei Zhao, and Liquan Shen. Co-saliency detection via co-salient object discovery and recovery. *SPL*, 2015. 6, 8

[50] Mikhail Zaslavskiy, Francis Bach, and Jean-Philippe Vert. A path following algorithm for the graph matching problem. *TPAMI*, 2008. 3

[51] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *CVPR*, 2020. 2, 8

[52] Dingwen Zhang, Huazhu Fu, Junwei Han, Ali Borji, and Xuelong Li. A review of co-saliency detection algorithms: Fundamentals, applications, and challenges. *TIST*, 2018. 2, 6

[53] Dingwen Zhang, Junwei Han, Jungong Han, and Ling Shao. Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining. *TNNLS*, 2016. 6, 8

[54] Dingwen Zhang, Junwei Han, Chao Li, and Jingdong Wang. Co-saliency detection via looking deep and wide. In *CVPR*, 2015. 2, 6

[55] Dingwen Zhang, Deyu Meng, and Junwei Han. Co-saliency detection via a self-paced multiple-instance learning framework. *TPAMI*, 2016. 2

[56] Kaihua Zhang, Jin Chen, Bo Liu, and Qingshan Liu. Deep object co-segmentation via spatial-semantic network modulation. In *AAAI*, 2020. 1, 6, 8

[57] Kaihua Zhang, Tengpeng Li, Bo Liu, and Qingshan Liu. Co-saliency detection via mask-guided fully convolutional networks with multi-scale label smoothing. In *CVPR*, 2019. 1, 2, 6, 8

[58] Kaihua Zhang, Tengpeng Li, Shiwen Shen, Bo Liu, and Qingshan Liu. Adaptive graph convolutional network with attention graph clustering for co-saliency detection. In *CVPR*, 2020. 1, 2, 6, 8

[59] Zhao Zhang, Wenda Jin, Jun Xu, and Ming-Ming Cheng. Gradient-induced co-saliency detection. In *ECCV*, 2020. 2, 6, 8

[60] Jiaxing Zhao, Jiang Jiang Liu, Deng Ping Fan, Yang Cao, Jufeng Yang, and Ming Ming Cheng. Egnet: Edge guidance network for salient object detection. In *ICCV*, 2019. 6, 8

[61] Qi Zhao, Zhi Yang, and Hai Tao. Differential earth mover's distance with its applications to visual tracking. *TPAMI*, 2008. 2