# Same-class Object Co-localization as Maximum Edge Weight Cliques

**Yi Fan**[1,2,3] and **Longin Jan Latecki**[4]
[1]School of Mathemetics and Statistics, Qiannan Normal University for Nationalities,
[2]Key Laboratory of Complex Systems and Intelligent Optimization of Guizhou Province,
Duyun, 558000, China
[3]Guilin University of Electronic Technology, Guilin 541004, China
[4]Temple University, Philadelphia, USA
yifan.sysu@gmail.com, latecki@temple.edu

## Abstract

The task of same-class object co-localization is of great importance. It allows a massive collection of images to be arranged in a classified way and helps searching engines to retrieve images of a given topic more efficiently. However, it can be a challenging problem due to viewpoint, occlusion, multiple same-class objects in a single image, considerable diversity in a certain class as well as time requirements. In this paper, we introduced our published papers which formulated this task as solving the maximum edge weight clique (MEWC) problem. More specifically we first adopted a list of deep learning techniques to obtain objects and similarity measures for constructing associated graphs, and then we called a local search solver to look for an MEWC which corresponds to a group of same-class objects. Experimental results not only show that our method outperforms state-of-the-arts but also confirms the individual impacts of our key components.

## 1 Introduction

Given a set of images, the task of same-class object co-localization, also known as image co-localization or common-object discovery [Tang *et al.*, 2014; Joulin *et al.*, 2014; Fan *et al.*, 2017; Rao *et al.*, 2019], is to simultaneously localize objects of the same class in each image. Co-localization can be important, since it allows a massive collection of images, like those from Facebook or Youtube, to be stored in a classified way, which in turn, helps searching engines to find images of a given topic more efficiently. This mimics a situation in smart phones, where copies or references of images automatically lie in different folders each of which corresponds to a certain person. What's more, similar approaches can be applied to a huge set of text documents as well.

In this paper we introduce a method to deal with the task of same-class object co-localization, which aims at locating objects of the same class in an image collection. By convention, we further require that at most one object can be obtained in each image. Co-localizing objects in unconstrained environments is challenging. In the real-world applications, objects of the same class may look different due to viewpoint, occlusion, deformation, illumination, etc. Besides, there could be considerable diversities even within the same object class. Take human beings for instance, they may differ from each other because of gender, age, costume, hair style or skin color. Also, there can be multiple same-class objects lying in the same set of images. In addition, efficiencies can sometimes be vital in time-sensitive applications such as in large collections of images or video streams.

The Maximum Edge Weight Clique (MEWC) problem is defined over a simple undirected graph $G = (V, E, w_{\mathcal{E}})$, where $V = \{v_1, \ldots, v_n\}$ is the vertex set, each edge $e \in E$ is a 2-element subset of $V$, and $w_{\mathcal{E}} : E \mapsto R_{\geq 0}$ is a weighting function on $E$. A clique $C$ is a subset of vertices in $G$ such that each pair of vertices in $C$ is connected. The MEWC problem is to find a clique $C$ which maximizes $\sum_{v_i, v_j \in C} w_{\mathcal{E}}(\{v_i, v_j\})$.

To achieve robust and efficient object co-localization, we formulated this task as solving the MEWC problem. More specifically, we utilized to deep learning to construct an associated graph, in which each vertex represents to a single object candidate generated from a given image collection, while the weight on an edge $e = \{u, v\}$ indicates how (visually) similar $u$'s and $v$'s corresponding object candidates are. To ensure that at most one object will be selected in any image, we further required edges as follows. Two vertices are connected by an edge if their corresponding object candidates are from different images, otherwise, they are disconnected. In this sense, any clique corresponds to a group of objects from different images and vice versa. Hence, we can locate a set of most mutually-similar objects by obtaining an MEWC in the associated graph, and each vertex in the MEWC is a localized same-class object across images (See Figure 1).

The main contributions of our work are as follows.

1. We adopted deep learning to formulate the task of same-class object co-localization as an MEWC problem in an associated graph, which in turn, provides an industrial benchmark for research and applications about MEWC algorithms.
2. We confirmed the effectiveness of local search in achieving high percentage of images with correct object co-localization.
3. We found that the Region Proposal Network (RPN)

Figure 1: Given a set of object candidates generated from an image collection (left), our goal is to find same-class objects by searching for a maximum edge weight clique in the associated graph. Each vertex in the clique (right) corresponds to a same-class object [Rao *et al.*, 2019].

[Ren *et al.*, 2015] effectively generates object candidates which are then re-ranked to improve robustness against background noises.

4. We trained a Triplet Network (TN) to obtain feature embeddings of object candidates, with the intention to construct a reliable affinity measure between the candidates.
5. Our method outperformed state-of-the-arts on both the PASCAL VOC 2007 image dataset [Everingham *et al.*, 2007] and the YouTube-Objects video dataset [Kalogeiton *et al.*, 2016].

## 2 Related Work

The problem of same-class object co-localization has been investigated extensively during the last decade. [Papazoglou and Ferrari, 2013] model this task as a foreground object mining problem, and they adopt Optical Flow and Gaussian Mixture models to accomplish the task. [Cho *et al.*, 2015] tackle this co-localization problem by a part-based region matching method and apply a probabilistic Hough transform to evaluate each candidate correspondence. [Joulin *et al.*, 2014] extend the approach in [Cho *et al.*, 2015] to co-localize objects in video frames, and they utilize a Frank-Wolfe algorithm to optimize their quadratic programming algorithm. [Zhang *et al.*, 2015] apply a part-based object detector as well as a motion aware region detector to generate object candidates, and they further formulate this problem as a joint assignment problem and then refine their solution by inferring shape likelihoods. [Kwak *et al.*, 2015] also focus on the problem of localizing dominant objects in videos, in which they apply an iterative process of detection and tracking. [Li *et al.*, 2016] devise an entropy-based objective function to learn a common object detector, and they address the task of co-localization with a Conditional Random Field (CRF) model. [Wei *et al.*, 2017] perform Principal Component Analysis (PCA) on the convolutional feature maps of all images, and locate the most correlated regions across images. [Wang *et al.*, 2017] use segmentations produced by Fully Convolutional Networks (FCN) as object candidates, and they formulate the task of same-class object co-localization as an $N$-Partite Graph Matching problem.

## 3 Modeling

Given a set of images $\mathcal{I}$, we apply deep learning methods to obtain a set of object candidates $\mathcal{B}$ from all images. To be specific, we let $\mathcal{B} = \bigcup_{I \in \mathcal{I}} \mathcal{P}(I) = \{\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n\}$, where $\mathcal{P}(I)$ is the set of object candidates extracted from image $I$, and $n$ is the total number of candidates generated from all images, so the size of $\mathcal{B}$ is $n$.

Intuitively, two objects are likely to be in the same class if they look similar, so in solving the problem of same-class object co-localization, we paid attention to visual similarity. Given two object candidates $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$, we use $s(\boldsymbol{b}_i, \boldsymbol{b}_j)$ to denote some certain similarity measure between them, then the task of same-class object co-localization can be formulated as searching for an optimal or near-optimal subset $\mathcal{B}^* \subset \mathcal{B}$ which maximizes

$$\sum_{\boldsymbol{b}_i, \boldsymbol{b}_j \in \mathcal{B}, \boldsymbol{b}_i \neq \boldsymbol{b}_j} s(\boldsymbol{b}_i, \boldsymbol{b}_j),$$

with the constraint that at most one object candidate can be selected from each image.

Then we turned the problem above into an MEWC problem as follows. We first constructed an associated graph $G$ with $n$ vertices, in which each vertex $v_i$ represents an object candidate $\boldsymbol{b}_i$.

To ensure that we will select at most one object candidate from each image, we required the edges in $G$ as follows. Two vertices $v_i$ and $v_j$ are connected by an edge if their corresponding object candidates $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ are from different images, otherwise, they are disconnected. Hence, if we obtain a clique $C$ from the associated graph $G$, any pair of vertices in $C$ must represent object candidates from different images (See the left picture in Figure 1).

To maximize mutual similarity between selected object candidates, we assigned each edge a positive weight as follows. An edge $e = \{v_i, v_j\}$ has a weight $w_{\mathcal{E}}(e) = s(\boldsymbol{b}_i, \boldsymbol{b}_j)$, where $\boldsymbol{b}_i$ and $\boldsymbol{b}_j$ are the corresponding object candidates of $v_i$ and $v_j$ respectively.

In this sense, the task of same-class object co-localization is formulated as the MEWC problem and the obtained clique contains objects that are probably in the same class (See the right picture in Figure 1).

## 4 A Detailed Approach

In this section, we introduce details about constructing associated graphs and local search for MEWC.

### 4.1 Choosing Object Candidates

As we see in Section 3, the vertices in our associated graph correspond to object candidates from all images. To improve

| MEWC Solvers | Aero | Bike | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Table | Dog | Horse | MBike | Person | Plant | Sheep | Sofa | Train | TV | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [Ma and Latecki, 2012] | 63.0 | 45.7 | 56.1 | 51.9 | 14.3 | 47.8 | 71.9 | 61.1 | 36.2 | **72.3** | **46.0** | 57.2 | 61.3 | 79.6 | 62.3 | 34.3 | 69.8 | 39.3 | 59.4 | 9.8 | 52.0 |
| [Wang *et al.*, 2016] | 61.3 | 68.3 | 61.2 | 48.1 | 16.8 | 67.7 | 76.9 | 58.5 | 41.8 | **72.3** | 24.5 | 63.9 | 68.3 | 75.5 | 69.0 | 28.6 | **76.0** | 47.2 | 62.1 | 68.4 | 57.8 |
| [Wang *et al.*, 2016]+BMS | 63.9 | 68.3 | 60.9 | 50.3 | 49.2 | 66.7 | 76.9 | 59.3 | 41.1 | **72.3** | 23.0 | **65.1** | 68.3 | 77.1 | **69.8** | 27.8 | 76.0 | 45.9 | 62.8 | 68.4 | 59.7 |
| [Adamczewski *et al.*, 2015] | 62.2 | **69.5** | **62.1** | 52.5 | 18.4 | **71.5** | 78.5 | 61.1 | **49.2** | 70.9 | 30.0 | 62.0 | 69.7 | 80.8 | 66.0 | **49.4** | 70.8 | 50.2 | 63.6 | 68.0 | 60.3 |
| [Fan *et al.*, 2017] | **64.7** | 58.4 | 60.3 | **54.1** | **52.0** | 71.0 | **79.2** | **63.8** | 43.1 | 71.6 | 40.5 | 64.4 | **72.1** | **84.9** | 69.5 | 45.3 | 75.0 | **51.1** | **66.3** | **71.1** | **62.9** |

Table 1: Co-localization CorLoc (%) of different MEWC solvers on the PASCAL07 dataset

| CNN Backbones | RPN Objectness | Object Proposal Re-ranking | Triplet Loss Fine-tuning |
|---|---|---|---|
| Pre-trained VGG-f Model | 31.2 | 53.8 | 59.3 |
| Pre-trained VGG-16 Model | 33.1 | 56.1 | 62.9 |

Table 2: Co-localization CorLoc (%) of different associated graph constructions on the PASCAL07 dataset

the quality of same-class object co-localization, we desired to cover as many foreground objects as possible. However, in solving the MEWC problem, the search space grows exponentially wrt. the number of vertices in the associated graph, i.e., the number of object candidates $|\mathcal{B}|$. Thus, it is vital to find a proper method to generate object candidates. Our main idea for choosing object candidates is as follows. (1) We adopted the Region Proposal Networks (RPN) [Ren *et al.*, 2015] to generate rectangular bounding boxes, each of which contains a single object candidate. (2) Then we applied Non-Maximum Suppression (NMS) to remove redundant boxes. (3) Last we chose top-$k$ scoring proposals in each image to construct our associated graph.

In choosing the top-$k$ scoring proposals, we considered two different proposal scoring measures. The first one is widely used and is based on the RPN objectness score of each object candidate. The second one is described as follows. Apart from object candidates, RPN also generates a vector of class likelihoods, i.e., a probability distribution over different classes, for each object candidate. Thus *we proposed to re-rank the object candidates according to the entropy of the class distribution*. Since the entropy is a measure of uncertainty, it serves a similar purpose as the objectness score but tend to be more accurate in this setting. Hence we can re-rank the raw RPN proposals according to the entropy, and selected the top-$k$ scoring boxes with low uncertainty as object candidates in each image.

## 4.2 Object Representation and Similarity Measure

As mentioned in Section 3, the edge weights in an associated graph represent visual similarity between object candidates. Thus, we needed a suitable way to accurately represent the object candidates and evaluate similarities between each pair of them. In this paper, we employed the Triplet Network framework [Hoffer and Ailon, 2015] to learn deep feature embeddings of the object candidates, because it makes similar objects closer to each other and dissimilar ones farther from each other in the specified metric space.

Suppose a pre-trained convolutional neural network (CNN) is selected to extract deep features $f(\boldsymbol{b};\boldsymbol{w})$ for each object candidate $\boldsymbol{b} \in \mathcal{B}$, where $\boldsymbol{w}$ is the set of parameters of the CNN. In our graph construction framework, a set of triplets will then be constructed for *fine-tuning* the parameters $\boldsymbol{w}$. Each triplet consists of a reference object $\boldsymbol{b_r}$, a positive object $\boldsymbol{b_p}$ and a negative object $\boldsymbol{b_n}$. More specifically, $\boldsymbol{b_r}$ and $\boldsymbol{b_p}$ are a pair of similar objects in the sense that they belong to the same category, while $\boldsymbol{b_r}$ and $\boldsymbol{b_n}$ are dissimilar which means the opposite. In this sense, the hinge loss of a triplet is defined as

$$l(\boldsymbol{b_r}, \boldsymbol{b_p}, \boldsymbol{b_n}) = \max\{0, \lambda + s(\boldsymbol{b_r}, \boldsymbol{b_n}) - s(\boldsymbol{b_r}, \boldsymbol{b_p})\}, \quad (1)$$

where $\lambda$ is a margin threshold indicating in which situation $l(\boldsymbol{b_r}, \boldsymbol{b_p}, \boldsymbol{b_n})$ will matter. More specifically, once $s(\boldsymbol{b_r}, \boldsymbol{b_p})$ is greater than $s(\boldsymbol{b_r}, \boldsymbol{b_n})$ by $\lambda$ or even more, $l(\boldsymbol{b_r}, \boldsymbol{b_p}, \boldsymbol{b_n})$ will make no contribution to our objective function (2).

To make similar objects closer and dissimilar ones farther, the Triplet Network learning process tries to find a set of optimal parameters $\boldsymbol{w}$ to minize the sum of the hinge loss

$$L(\mathcal{T}) = \sum_{\boldsymbol{b_r}, \boldsymbol{b_p}, \boldsymbol{b_n} \in \mathcal{T}} l(\boldsymbol{b_r}, \boldsymbol{b_p}, \boldsymbol{b_n}) \quad (2)$$

over a training set of triplets $\mathcal{T}$.

After the optimization process above, we obtained deep features $f(\boldsymbol{b};\boldsymbol{w})$ which helped produce better similarity measures. In our work, we employed the cosine similarity between two CNN feature vectors $f(\boldsymbol{b_i};\boldsymbol{w})$ and $f(\boldsymbol{b_j};\boldsymbol{w})$ as the visual similarity $s(\boldsymbol{b_i}, \boldsymbol{b_j})$ in Equation (1) above, namely

$$s(\boldsymbol{b_i}, \boldsymbol{b_j}) = \frac{f(\boldsymbol{b_i};\boldsymbol{w})^{\mathsf{T}} f(\boldsymbol{b_j};\boldsymbol{w})}{\|f(\boldsymbol{b_i};\boldsymbol{w})\| \cdot \|f(\boldsymbol{b_j};\boldsymbol{w})\|},$$

which is simple, neatly bounded and parameter free.

## 4.3 Local Search for MEWC

With an associated graph constructed above, we called a local search algorithm named CERS in [Fan *et al.*, 2017] to find an optimal or near-optimal solution, in order to obtain same-class objects across different images. This algorithm restarts whenever it revisits a local optimum, and it exploits a hash table to implement this idea approximately. To be specific, it utilize a hash function as

$$hash(\mathcal{B}_c) = \left( \sum_{\boldsymbol{b}_i \in \mathcal{B}_c} 2^i \right) \mod p, \quad (3)$$

where $i \in \{1, \cdots, n\}$ is the index of $\boldsymbol{b}_i$ in the entire object candidate set $\mathcal{B}$ and $p$ is a prime. If $p$ is sufficiently large, the

| Method | Aero | Bike | Bird | Boat | Bottle | Bus | Car | Cat | Chair | Cow | Table | Dog | Horse | MBike | Person | Plant | Sheep | Sofa | Train | TV | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [Joulin *et al.*, 2014] | 32.8 | 17.3 | 20.9 | 18.2 | 4.5 | 26.9 | 32.7 | 41.0 | 5.8 | 29.1 | 34.5 | 31.6 | 26.1 | 40.4 | 17.9 | 11.8 | 25.0 | 27.5 | 35.6 | 12.1 | 24.6 |
| [Cho *et al.*, 2015] | 50.3 | 42.8 | 30.0 | 18.5 | 4.0 | 62.3 | **64.5** | 42.5 | 8.6 | **49.0** | 12.2 | 44.0 | 64.1 | 57.2 | 15.3 | 9.4 | 30.9 | 34.0 | 61.6 | 31.5 | 36.6 |
| [Li *et al.*, 2016] | 73.1 | 45.0 | 43.4 | 27.7 | 6.8 | 53.3 | 58.3 | 45.0 | 6.2 | 48.0 | 14.3 | 47.3 | 69.4 | 66.8 | 24.3 | 12.8 | **51.5** | 25.5 | 65.2 | 16.8 | 40.0 |
| [Wang *et al.*, 2015] | 37.7 | 58.8 | 39.0 | 4.7 | 4.0 | 48.4 | 70.0 | 63.7 | 9.0 | 54.2 | 33.3 | 37.4 | 61.6 | 57.6 | 30.1 | 31.7 | 32.4 | 52.8 | 49.0 | 27.8 | 40.2 |
| [Bilen *et al.*, 2015] | 66.4 | 59.3 | 42.7 | 20.4 | 21.3 | 63.4 | 74.3 | 59.6 | 21.1 | 58.2 | 14.0 | 38.5 | 49.5 | 60.0 | 19.8 | 39.2 | 41.7 | 30.1 | 50.2 | 44.1 | 43.7 |
| [Ren *et al.*, 2016] | **79.2** | 56.9 | 46.0 | 12.2 | 15.7 | 58.4 | 71.4 | 48.6 | 7.2 | **69.9** | 16.7 | 47.4 | 44.2 | 75.5 | **41.2** | 39.6 | 47.4 | 32.2 | 49.8 | 18.6 | 43.9 |
| [Wei *et al.*, 2017] | 67.3 | 63.3 | 61.3 | 22.7 | 8.5 | 64.8 | 57.0 | 80.5 | 9.4 | 49.0 | 22.5 | 72.6 | 73.8 | 69.0 | 7.2 | 15.0 | 35.3 | 54.7 | 75.0 | 29.4 | 46.9 |
| [Wang *et al.*, 2017] | 80.1 | 63.9 | 51.5 | 4.9 | 21.0 | 55.7 | 74.2 | 43.5 | 26.2 | 53.4 | 16.3 | 56.7 | 58.3 | 69.5 | 14.1 | 38.3 | 58.8 | 47.2 | 49.1 | 60.9 | 48.5 |
| [Cinbis *et al.*, 2016] | 67.1 | 66.1 | 49.8 | 34.5 | 23.3 | 68.9 | **83.5** | 44.1 | 27.7 | 71.8 | 49.0 | 48.0 | 65.2 | 79.3 | 37.4 | 42.9 | 65.2 | 51.9 | 62.8 | 46.2 | 54.2 |
| [Rochan and Wang, 2015] | 78.5 | 63.3 | 66.3 | 56.3 | 19.6 | 82.2 | 74.7 | 69.1 | 22.4 | 72.3 | 31.0 | 62.9 | 74.9 | 78.3 | 48.6 | 29.3 | 64.5 | 36.2 | 75.8 | 69.5 | 58.8 |
| Ours with VGG-f | 59.7 | 67.1 | 60.3 | 46.4 | 51.2 | 68.8 | 75.9 | 57.9 | 40.4 | 77.3 | 21.5 | 64.6 | 65.2 | 74.7 | 67.3 | 41.6 | 77.1 | 48.0 | 60.9 | 60.9 | 59.3 |
| Ours with VGG-16 | **64.7** | 58.4 | 60.3 | **54.1** | **52.0** | 71.0 | **79.2** | **63.8** | 43.1 | 71.6 | 40.5 | 64.4 | **72.1** | **84.9** | 69.5 | 45.3 | 75.0 | **51.1** | **66.3** | **71.1** | **62.9** |

Table 3: Co-localization CorLoc (%) of different methods on the PASCAL07 dataset

| Method | Aeroplane | Bird | Boat | Car | Cat | Cow | Dog | Horse | MotorBike | Train | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [Prest *et al.*, 2012] | 51.7 | 17.5 | 34.4 | 34.7 | 22.3 | 17.9 | 13.5 | 26.7 | 41.2 | 25.0 | 28.5 |
| [Joulin *et al.*, 2014] | 25.1 | 31.2 | 27.8 | 38.5 | 41.2 | 28.4 | 33.9 | 35.6 | 23.1 | 25.0 | 30.9 |
| [Papazoglou and Ferrari, 2013] | 65.4 | 67.3 | 38.9 | 65.2 | 46.3 | 40.2 | 65.3 | 48.4 | 39.0 | 25.0 | 50.1 |
| [Zhang *et al.*, 2015] | **75.8** | 60.8 | 43.7 | 71.1 | 46.5 | 54.6 | 55.5 | 54.9 | 42.4 | 35.8 | 54.1 |
| [Rochan and Wang, 2015] | 56.0 | 30.1 | 39.6 | **85.7** | 24.7 | **87.8** | 55.6 | 60.2 | 61.8 | 51.7 | 55.3 |
| Ours with VGG-f | 48.5 | **74.4** | 52.8 | 61.6 | **59.4** | 69.3 | **71.4** | 68.5 | 73.6 | 43.0 | 62.3 |
| Ours with VGG-16 | 44.3 | 68.6 | **56.7** | 63.5 | 50.0 | 70.7 | 71.2 | **75.9** | **73.8** | **55.5** | **63.0** |

Table 4: Co-localization CorLoc (%) of different methods on the YouTube-Objects dataset

probability of hash collision is negligible, so we simply set $p$ according to the memory available in the running environment.

We realized that [Chu *et al.*, 2020] developed a local search algorithm which is able to solve the MEWC problem here, yet they did not test their algorithm in this application domain.

# 5 Experiments

To evaluate the performance of our method, we conducted experiments on the PASCAL VOC 2007 image dataset [Everingham *et al.*, 2007] and the YouTube-Objects video dataset [Kalogeiton *et al.*, 2016]. For evaluation, we adopted the standard PASCAL criterion Intersection over Union (IoU), to be specific, a predicted bounding box $\boldsymbol{b}^p$ is correct if

$$IoU(\boldsymbol{b}^p, \boldsymbol{b}^{gt}) = \frac{area(\boldsymbol{b}^p \cap \boldsymbol{b}^{gt})}{area(\boldsymbol{b}^p \cup \boldsymbol{b}^{gt})} > 0.5,$$

where $\boldsymbol{b}^{gt}$ is a ground-truth annotation of the bounding box. Finally, we utilized the percentage of images with correct object co-localization (CorLoc) [Li *et al.*, 2016] as the evaluation measure.

## 5.1 Experimental Setup

We adopted GPU to speed up deep learning processes and exploited pre-trained networks for higher accuracy and shorter training-validation time. Also we fine-tuned the pre-trained networks by Triplet networks for better accuracies in the task of same-class object co-localization. Last we implemented the high level part of our system in Matlab and it called the local search solver CERS to find optimum or near-optimum solutions to the MEWC problem.

**Running Environments**

We carried out experiments on a desktop with two Intel(R) Core(TM) i7 CPUs (2.80GHz) and 64 GB memory, and we also exploited a GeForce GTX Titan X GPU to train and test related deep neural networks.

**Pre-trained Networks**

We built the RPN and the Triplet Network in our method upon the VGG-f model [Simonyan and Zisserman, 2015] as well as the VGG-16 model [Chatfield *et al.*, 2014]. Compared to the VGG-16 model, the structure of the VGG-f model is much simpler and thus more computationally efficient. The VGG-f and VGG-16 models are pre-trained on the ImageNet dataset [Russakovsky *et al.*, 2015] and fine-tuned on the Microsoft COCO dataset [Lin *et al.*, 2014]. All parameters were fixed the same in the experiments unless explicitly stated.

**Implementations**

We programmed the high level part of our system in MATLAB with some utilities written as MEX fles. As to the phase of constructing associated graphs, we utilized the deep learning frameworks Caffe and MatConvNet as carriers for building the RPN and the Triplet Network. When solving the MEWC problem, we called the CERS algorithm which was implemented in C/C++.

**Parameter Settings**

We adopted default parameters to learn RPN and generated object candidates and also used a threshold of 0.5 for the NMS to remove redundant object proposals. In each image, we selected the best $k = 20$ object candidates. We set $\lambda = 0.25$ in the hinge loss function (1) of a single triplet. As to the local search algorithm CERS for MEWC, it has a parameter $p$ in Equation (3) which is required to be a prime. We set it to $10^9 + 7$ and the hash table in it consumed memory of around 1 GB.

**Evaluation Protocol**

As randomness may exist in different methods, in each table we tested each method 10 times with different seeds on the

dataset and reported the average unless explicitly stated.

### 5.2 Performances on the PASCAL Dataset

We adopted the PASCAL VOC 2007 dataset [Everingham *et al.*, 2007] to evaluate the performances of object co-localization in images. This dataset is split as a training-validation set and a test set, each with about 5,000 images in 20 classes (See Table 1). We followed [Joulin *et al.*, 2014] to construct a collection of images for object co-localization from the training-validation set and denoted it as PASCAL07. This is fine in our framework, since our RPN and Triplet networks are trained on the ImageNet and COCO datesets instead of this PASCAL07. In other words, we kept the test set secret during our training and validation processes.

**Individual Impacts of MEWC Solvers**

To confirm the individual impact of our selected MEWC solver, i.e., CERS, we first replaced it with other state-of-the-art MEWC algorithms and performed experiments on the same set of graph instances. These instances were generated from PASCAL07 based on the VGG-16 model. Since PASCAL07 has images from 20 different classes, we constructed 20 different graphs, one graph for each image class. The details about associated graph constructions can be found in Sections 4.1 and 4.2. As a result, the average number of vertices in the constructed graphs is 6081.67, and the average number of edges is $2.16 \times 10^7$, thus the average density of the graphs is 0.9962.

In Table 1, we presented co-localization accuracies (CorLoc) of different MEWC solvers when embedded in our framework. We showed results of each solver in a single row, including results for the 20 classes in PASCAL07 and also the average result over those 20 classes.

The method in [Ma and Latecki, 2012] solves the MEWC problem in the relaxed continuous domain and it proposes a modified Frank-Wolfe algorithm to tackle this problem. Note that in [Wang *et al.*, 2016], there are two versions of their solver, LSCC and LSCC+BMS, therefore in Table 1, we presented their results in two rows. CERS [Fan *et al.*, 2017] inherits much from MN/TS [Wu *et al.*, 2012], LSCC [Wang *et al.*, 2016] and LMY-GRS [Fan *et al.*, 2016], and they four form a family of local search solvers. The algorithm TBMA [Adamczewski *et al.*, 2015] also adopts local search to solve the MEWC problem directly in the discrete domain like the family above, and it restarts if the quality of candidate solutions has not been improved for a specified number of steps.

From the last column in Table 1, we found that CERS is the most suitable solver in our framework.

**Individual Impacts of Associated Graph Constructions**

To evaluate how our algorithm depended on the performance of the deep learning method used, we proposed several variants by disabling one or more processes. The detailed results are shown in Table 2.

We utilized different CNN models to extract object candidate features, and then applied the cosine similarity on these features. In Section 4.1, we mentioned that there are two different proposal scoring measures. The first one is to use the RPN objectness score. The second one is to utilize entropy-based re-ranking. Comparing Columns 2 and 3, we find that the second ranking criteria significantly outperformed the first one. In addition, Column 4 and those previous ones presents the contribution of the Triplet Network learning framework. Last, the rows in this table show the superiority of the VGG-16 model over the VGG-f model in terms of accuracies.

In a word, the experiments above validated that the performance of object co-localization benefits from proper choices of the object candidate generation and the feature embedding scheme.

**Effectiveness of Our Whole Method**

We reported the accuracy of different object co-localization methods on the PASCAL07 dataset in Table 3 which follows the same presentation protocol as Table 1.

The results of the competitors were directly taken from the corresponding literature, and the most important competitors were briefly introduced in Section 2. Among those methods which adopt deep CNN features as visual descriptors [Li *et al.*, 2016; Wang *et al.*, 2015; Bilen *et al.*, 2015; Ren *et al.*, 2016; Wei *et al.*, 2017; Cinbis *et al.*, 2016; Rochan and Wang, 2015], our method demonstrates superior performances. The experiments testified the effectiveness of our whole object co-localization method in images.

Last we compared our formulation here with an earlier one in [Fan *et al.*, 2017] which also adopts CERS for MEWC. Experimental results on PASCAL07 showed that their accuracy rate is 57.2% which is lower than that of each of ours.

### 5.3 Performances on the YouTube-Objects Dataset

We used the Youtube-Objects dataset [Kalogeiton *et al.*, 2016] for object co-localization in videos. It contains videos collected from YouTube with 10 object classes. There are about 570,000 frames with 1,407 annotations in the first version of the dataset [Prest *et al.*, 2012]. To our best, it is the largest available video dataset with bounding-box annotations on multiple classes. To avoid possible confusion when applying different video decoders, we used the individual video frames after decompression in our experiments. Moreover, we only performed object co-localization on video frames with ground-truth annotations, following the practice in [Joulin *et al.*, 2014]. Furthermore, we made no use of additional spatial-temporal information. The Youtube-Objects dataset comes with the test videos divided in 10 classes according to which dominant object occurs most in them. Hence, we constructed 10 different graphs for this dataset.

Table 4 follows the same presentation protocol as Tables 1 and 3, and it summarizes the co-localization accuracy of different methods on the YouTube-Objects dataset. Among all the methods, [Zhang *et al.*, 2015; Rochan and Wang, 2015] also utilized deep networks for visual representation. The results justified that the proposed object co-localization method is also effective for mining same-class objects in videos.

## 6 Conclusions and Future Works

In this paper, we addressed the task of same-class object co-localization which aims at finding a group of maximum mutually-similar objects. We formulated this problems as the MEWC problem and adopted a local search solver to search for an optimal or near-optimal solution. Experimental results

not only shows that our whole method significantly outperformed state-of-the-arts, but also confirms the individual impacts of our key components.

However, in reality there can be more than one object in a single image that belongs to a certain class. Also there can be no objects in a single image which belongs to a particular class. Hence, we will extend the task in this paper to cover these cases in future.

## References


[Adamczewski *et al.*, 2015] Kamil Adamczewski, Yumin Suh, and Kyoung Mu Lee. Discrete tabu search for graph matching. In *Proceedings of the IEEE international conference on computer vision*, pages 109–117, 2015.

[Bilen *et al.*, 2015] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly supervised object detection with convex clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1081–1089, 2015.

[Chatfield *et al.*, 2014] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In Michel François Valstar, Andrew P. French, and Tony P. Pridmore, editors, *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*. BMVA Press, 2014.

[Cho *et al.*, 2015] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1201–1210, 2015.

[Chu *et al.*, 2020] Yi Chu, Boxiao Liu, Shaowei Cai, Chuan Luo, and Haihang You. An efficient local search algorithm for solving maximum edge weight clique problem in large graphs. *Journal of Combinatorial Optimization*, 39(4):933–954, 2020.

[Cinbis *et al.*, 2016] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2016.

[Everingham *et al.*, 2007] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.

[Fan *et al.*, 2016] Yi Fan, Chengqian Li, Zongjie Ma, Abdul Sattar, Lian Wen, and Kaile Su. Local search for maximum vertex weight clique on large sparse graphs with efficient data structures. In *AI 2016: Advances in Artificial Intelligence - 29th Australasian Joint Conference, Hobart, Australia, December 4-8, 2016. Proceedings. To appear.*, 2016.

[Fan *et al.*, 2017] Yi Fan, Zongjie Ma, Kaile Su, Chengqian Li, Cong Rao, Ren-Hau Liu, and Longin Jan Latecki. Efficient local search for maximum weight cliques in large graphs. In *29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017*, pages 1099–1104. IEEE Computer Society, 2017.

[Hoffer and Ailon, 2015] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In Aasa Feragen, Marcello Pelillo, and Marco Loog, editors, *Similarity-Based Pattern Recognition - Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015, Proceedings*, volume 9370 of *Lecture Notes in Computer Science*, pages 84–92. Springer, 2015.

[Joulin *et al.*, 2014] Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *Computer Vision–ECCV*, pages 253–268. Springer, 2014.

[Kalogeiton *et al.*, 2016] Vicky Kalogeiton, Vittorio Ferrari, and Cordelia Schmid. Analysing domain shift factors between videos and images for object detection. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2327–2334, 2016.

[Kwak *et al.*, 2015] Suha Kwak, Minsu Cho, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Unsupervised object discovery and tracking in video collections. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 3173–3181. IEEE Computer Society, 2015.

[Li *et al.*, 2016] Yao Li, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Image co-localization by mimicking a good detector's confidence score distribution. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II*, volume 9906 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2016.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

[Ma and Latecki, 2012] Tianyang Ma and Longin Jan Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 670–677. IEEE, 2012.

[Papazoglou and Ferrari, 2013] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE international conference on computer vision*, pages 1777–1784, 2013.

[Prest *et al.*, 2012] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *2012 IEEE Conference on Computer Vision and*

*Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3282–3289. IEEE Computer Society, 2012.

[Rao *et al.*, 2019] Cong Rao, Yi Fan, Kaile Su, and Longin Jan Latecki. Common object discovery as local search for maximum weight cliques in a global object similarity graph. In Michel Couprie, Jean Cousty, Yukiko Kenmochi, and Nabil H. Mustafa, editors, *Discrete Geometry for Computer Imagery - 21st IAPR International Conference, DGCI 2019, Marne-la-Vallée, France, March 26-28, 2019, Proceedings*, volume 11414 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2019.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[Ren *et al.*, 2016] Weiqiang Ren, Kaiqi Huang, Dacheng Tao, and Tieniu Tan. Weakly supervised large scale object localization with multiple instance learning and bag splitting. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 38(2):405–416, 2016.

[Rochan and Wang, 2015] Mrigank Rochan and Yang Wang. Weakly supervised localization of novel objects using appearance transfer. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4315–4324. IEEE Computer Society, 2015.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Tang *et al.*, 2014] Ke Tang, Armand Joulin, Li-Jia Li, and Li Fei-Fei. Co-localization in real-world images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1464–1471, 2014.

[Wang *et al.*, 2015] Xinggang Wang, Zhuotun Zhu, Cong Yao, and Xiang Bai. Relaxed multiple-instance svm with application to object discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1224–1232, 2015.

[Wang *et al.*, 2016] Yiyuan Wang, Shaowei Cai, and Minghao Yin. Two efficient local search algorithms for maximum weight clique problem. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 805–811, 2016.

[Wang *et al.*, 2017] Chuan Wang, Hua Zhang, Liang Yang, Xiaochun Cao, and Hongkai Xiong. Multiple semantic matching on augmented $n$ -partite graph for object co-segmentation. *IEEE Transactions on Image Processing*, 26(12):5825–5839, 2017.

[Wei *et al.*, 2017] Xiu-Shen Wei, Chen-Lin Zhang, Yao Li, Chen-Wei Xie, Jianxin Wu, Chunhua Shen, and Zhi-Hua Zhou. Deep descriptor transforming for image co-localization. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3048–3054, 2017.

[Wu *et al.*, 2012] Qinghua Wu, Jin-Kao Hao, and Fred Glover. Multi-neighborhood tabu search for the maximum weight clique problem. *Annals OR*, 196(1):611–634, 2012.

[Zhang *et al.*, 2015] Yu Zhang, Xiaowu Chen, Jia Li, Chen Wang, and Changqun Xia. Semantic object segmentation via detection in weakly labeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3641–3649, 2015.