CoLo-CAM: Class Activation Mapping for Object Co-Localization in Weakly-Labeled Unconstrained Videos

Soufiane Belharbi[1], **Shakeeb Murtaza**[1], **Marco Pedersoli**[1], **Ismail Ben Ayed**[1], **Luke McCaffrey**[2], **and Eric Granger**[1]
[1] LIVIA, Dept. of Systems Engineering, ÉTS, Montreal, Canada
[2] Goodman Cancer Research Centre, Dept. of Oncology, McGill University, Montreal, Canada
soufiane.belharbi.1@ens.etsmtl.ca

Abstract

Weakly-supervised video object localization (WSVOL) methods often rely on visual and motion cues only, making them susceptible to inaccurate localization. Recently, discriminative models via a temporal class activation mapping (CAM) method have been explored. Although results are promising, objects are assumed to have minimal movement leading to degradation in performance for relatively long-term dependencies. In this paper, a novel CoLo-CAM method for object localization is proposed to leverage spatiotemporal information in activation maps without any assumptions about object movement. Over a given sequence of frames, explicit joint learning of localization is produced across these maps based on color cues, by assuming an object has similar color across frames. The CAMs' activations are constrained to activate similarly over pixels with similar colors, achieving co-localization. This joint learning creates direct communication among pixels across all image locations, and over all frames, allowing for transfer, aggregation, and correction of learned localization. This is achieved by minimizing a color term of a CRF loss over joint images/maps. In addition to our multi-frame constraint, we impose per-frame local constraints including pseudo-labels, and CRF loss in combination with a global size constraint to improve per-frame localization. Empirical experiments[1] on two challenging datasets for unconstrained videos, YouTube-Objects, show the merits of our method, and its robustness to long-term dependencies, leading to new state-of-the-art localization performance.

Keywords: Convolutional Neural Networks, Weakly-Supervised Video Object Localization, Unconstrained Videos, Class Activation Maps (CAMs).

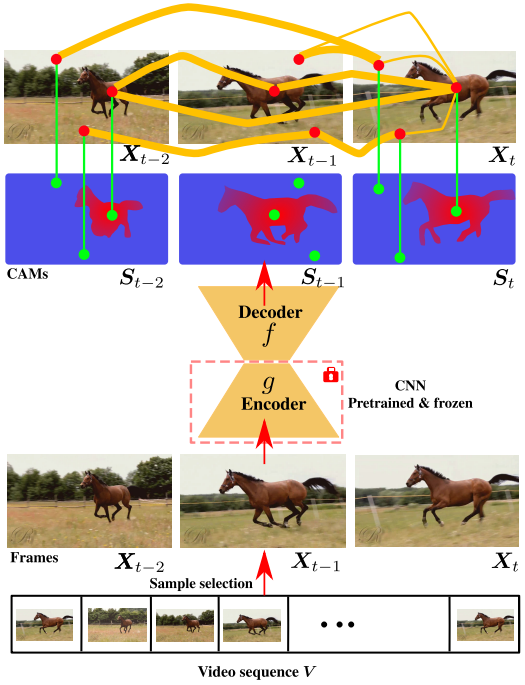


Figure 1: Illustration of the multi-frame training using our CoLo-CAM method with $n = 3$ frames. Each pixel (dot), is connected (orange line) to every pixel across the 3 frames to measure color similarity (connection thickness indicates similarity strength). CAM locations at pixels with similar colors are constrained to have similar activations (green lines are for alignment). For simplicity, per-frame terms are not visualized, and only a few pixel interconnections are shown. Notation and per-frame and multi-frame terms are described in Section 3.

1 Introduction

The recent progress of online multi-media platforms such as YouTube has provided easy access to large amounts of videos (Shao et al., 2022; Tang et al., 2013). Consequently, it is important to design techniques for automated video analysis. In this work, we focus on the task of object localization in videos, which plays a critical role in the understanding of video content, and the improvement of downstream tasks like video summarization (Zhang et al., 2017), event detection (Chang et al., 2016), video object detection (Chen et al., 2020; Han et al., 2020; Shao et al., 2022), facial expression recognition (Xue et al., 2022; Zhang et al., 2022), and visual object tracking (Bergmann et al., 2019; Luo et al., 2021). Unfortunately, the annotations required for fully supervised localization, *i.e.*, bounding boxes for each frame, come at a high cost for videos because of the large number of frames.

[1]Code: https://github.com/sbelharbi/colo-cam.

As an alternative, weak supervision using a video tag[2] is currently the most common annotation for weakly-supervised object localization in videos (Jerripothula et al., 2016; Tsai et al., 2016) and still images (Choe et al., 2020; Rony et al., 2023). Video tags often describe the main object appearing in the video without spatiotemporal information. However, individual frames do not necessarily contain the labeled class object, leading to uncertain annotations at the frame level. Unconstrained videos are often captured in the wild, with varying quality, moving objects and cameras, changing viewpoints and illumination, and decoding artifacts, and localizing objects in such videos is a challenging task. The weakly-supervised video object localization task (WSVOL) (Joulin et al., 2014; Koh et al., 2016; Kwak et al., 2015; Prest et al., 2012; Rochan et al., 2016; Zhang et al., 2020b) aims to localize objects spatially and temporally by predicting a bounding box for each frame, using weak supervision such as video tags. Other techniques perform weakly supervised object segmentation, then produce a bounding box via post-processing (Belharbi et al., 2023; Croitoru et al., 2019; Fu et al., 2014; Haller and Leordeanu, 2017; Liu et al., 2014; Tsai et al., 2016; Tokmakov et al., 2016; Umer et al., 2021; Yan et al., 2017; Zhang et al., 2014).

State-of-art WSVOL methods rely often all follow the same strategy. Initial segments/proposals are generated based on visual or motion cues, which are then used to identify and refine object locations through post-processing (Hartmann et al., 2012; Kwak et al., 2015; Prest et al., 2012; Tang et al., 2013; Tokmakov et al., 2016; Xu et al., 2012; Yan et al., 2017; Zhang et al., 2020b). This process is optimized under visual appearance and motion constraints to ensure consistency. Other methods rely on co-localization/co-segmentation across videos/images via graphs that allow for a larger view and interactions between segments (Chen et al., 2012; Fu et al., 2014; Jerripothula et al., 2016; Joulin et al., 2014; Tsai et al., 2016; Zhang et al., 2014). Despite their success, these methods have several limitations. Relying on multi-sequential training stages that are not end-to-end, makes them vulnerable to sub-optimal solutions. Often, localization is formulated as a solution to an optimization problem over one or more videos of the same class, and therefore requires a per-class/per-video model. This makes the inference time, and deployment expensive and impractical for real-world applications, in addition to a challenging scaling to large numbers of classes. Moreover, these methods start with initial proposals estimated in an unsupervised fashion (without the explicit use of video tags), and by relying on visual and motion cues such as optical flow (Lee et al., 2011; Sundaram et al., 2010). WSVOL methods are therefore vulnerable to inaccurate localization, especially with unconstrained videos, when motion cues are less reliable due to the movement of objects and cameras.

Recently, a discriminative multi-class DL model for WSVOL has been proposed (Belharbi et al., 2023). The authors consider using Class Activation Map (CAM)-based methods, which have been successful for weakly-supervised object localization (WSOL) on still images (Choe et al., 2020; Rony et al., 2023). Using only global image class labels, a CAM-based method allows training a DL model to classify an image, and localize the corresponding object via a CAM. Image regions corresponding to strong CAM activations indicate the potential presence of an object making it suitable for object localization with low-cost annotations (Oquab et al., 2015). However, these methods are not equipped to explicitly leverage temporal information in videos to improve localization. Therefore, the authors in (Belharbi et al., 2023) propose a temporal CAM-based method that performs spatiotemporal max-pooling to aggregate regions of interest (ROIs) across multiple CAMs from consecutive frames. Although this method can effectively exploit the short-term spatiotemporal dependencies in videos, it is vulnerable to the aggregation of inaccurate activations due to object movement. This issue is illustrated in their results with relatively long-term temporal dependencies (greater than frames), where the localization accuracy decreases rapidly.

To alleviate this issue while still benefiting from CAM methods, we introduce the CoLo-CAM method to leverage spatiotemporal information. In particular, a training framework with explicit joint learning of the CAM across multiple frames is considered. Using a *color* cue, we constrain CAMs extracted from a sequence of frames to be consistent by pushing them to activate similarly over pixels with a similar color. We rely on the assumption that objects in nearby frames appear similar, which is a fair assumption in videos. Unlike (Belharbi et al., 2023), we do not assume a minimal displacement of objects. Rather, they are allowed to be anywhere in the image, making our method more flexible. Our multi-frame optimization method is achieved via a *color* term of a CRF loss (Tang et al., 2018) applied simultaneously over all images/CAMs, thereby allowing for interconnection among all their pixels, and performing explicit co-localization. This opens an explicit communication channel between CAMs, allowing for the transfer, aggregation, and correction of learned knowledge. Similarly to (Belharbi et al., 2023), we use a U-Net style architecture (Ronneberger et al., 2015) that simultaneously classifies an image and locates the corresponding object. In addition to our new multi-frame constraint, we use per-frame local constraints including pseudo-labels and CRF loss (Belharbi et al., 2023; Tang et al., 2018), in combination with a global size constraint. This has been shown to improve per-frame localization (Belharbi et al., 2023). Our total training loss is composed of per-frame and multi-frame terms which

[2] An object class label, *i.e.*, video tag, is defined as weak annotation associated with the entire video.

are optimized simultaneously via standard Stochastic Gradient Descent (SGD). At inference, our method operates at a single frame level by simply performing a forward pass in the model, allowing a fast inference time.

Our main contributions are summarized as follows:

(1) We propose a novel CAM-based method called CoLo-CAM for the WSVOL task. Using a color cue, we explicitly constrain CAMs from a sequence of image frames to be consistent by activating similarly over pixels with similar color, achieving co-localization. This joint learning creates a direct communication between pixels across all image locations over all frames, allowing the transfer, aggregation, and correction of learned localization. This is achieved by minimizing a color term of a CRF loss (Tang et al., 2018) over joint images/CAMs. Unlike the recent CAM-based method (Belharbi et al., 2023), our temporal term does not assume the same object location in different frames but rather, allows the object to be situated anywhere, making it more flexible and robust to movements. Our multi-frame term is optimized jointly with other per-frame terms to boost localization performance.

(2) Our empirical experiments on two challenging public datasets of unconstrained videos, YouTube-Objects v1.0 (Prest et al., 2012) and v2.2 (Kalogeiton et al., 2016), show the merits of our method and its robustness to long dependency. This allows for achieving new state-of-the-art localization performance. Several ablations are provided. Our results also confirm the potential benefit of discriminative learning in WSVOL task.

2 Related Work

This section reviews weakly-supervised methods for localization/segmentation and co-localization/co-segmentation in videos. Additionally, CAM-based methods for WSOL on still images are discussed, as they relate to WSVOL.

Localization. Different methods exploit potential proposals as pseudo-supervision for localization (Prest et al., 2012; Zhang et al., 2020b). In (Prest et al., 2012), class-specific detectors are proposed. First, segments of coherent motion (Brox and Malik, 2010) are extracted, then a spatiotemporal bounding box is fitted to each segment forming tubes. A single tube is jointly selected by minimizing energy based on the similarity between tubes, visual homogeneity over time, and the likelihood to contain an object. The selected tubes are used as box supervision to train a per-class detector (Felzenszwalb et al., 2010). In (Zhang et al., 2020b), a DL model is trained using noisy supervision to perform segmentation and localization. Given a set of videos from the same class, bounding boxes and segmentation proposals are estimated using advanced optical flow (Lee et al., 2011). Other methods seek to discover a prominent object (foreground) in (a) video(s) (Koh et al., 2016; Rochan et al., 2016; Kwak et al., 2015). Then, similar regions are localized and refined using visual appearance and motion consistency. In (Rochan et al., 2016), box proposals are generated in a video, and only the relevant ones are retained for building an object appearance model. Maximum a posteriori inference over an undirected chain graphical model is employed to enforce the temporal appearance consistency between adjacent frames in the same video. The method in (Kwak et al., 2015) leverages two simultaneous and complementary processes for object localization: the discovery of similar objects in different videos, and the tracking of prominent regions in individual videos. Region proposals (Manen et al., 2013), in addition to appearance and motion confidence, are used to determine foreground objects (Brox and Malik, 2010), and to maintain temporal consistency between consecutive frames.

Segmentation. Independent spatiotemporal segments are first extracted (Hartmann et al., 2012; Tang et al., 2013; Xu et al., 2012; Yan et al., 2017) via unsupervised methods (Banica et al., 2013; Xu et al., 2012) or with a proposals generator (Zhang et al., 2015) using pretrained detectors (Zhang et al., 2015). Then, using multiple properties such as visual appearance, and motion cues, these segments are labeled, while preserving temporal consistency. Graphs, such as Conditional Random Field (CRF) and GrabCut approaches (Rother et al., 2004), are often employed to ensure consistency. In (Liu et al., 2014) deal with multi-class video segmentation is proposed, where a nearest neighbor-based label transfer between videos is exploited. First, videos are first segmented into spatiotemporal supervoxels (Xu et al., 2012), and represented by high-dimensional feature vectors using color, texture, patterns, and motion. These features are compressed using binary hashing for semantic similarity measures. A label transfer algorithm is designed using a graph that fosters label smoothness between spatiotemporal adjacent supervoxels in the same video, and between visually similar supervoxels across videos. In M-CNN (Tokmakov et al., 2016), a fully convolutional network (FCN) is combined with motion cues to estimate pseudo-labels. Using motion segmentation, a Gaussian mixture model is exploited to estimate foreground appearance potentials (Papazoglou and Ferrari, 2013). Combined with the FCN prediction, these potentials are used to yield better segmentation pseudo-labels via a GrabCut-like method (Rother et al., 2004), and these are then used to fit the FCN. The previously mentioned segmentation methods use video tags to cluster videos with the same class. While other methods do not rely on such supervision, the process is generally similar. Foreground regions are first estimated (Croitoru et al., 2019; Haller and Leordeanu, 2017; Papazoglou and Ferrari, 2013; Umer et al., 2021) using motion cues (Sundaram et al., 2010), or PCA, or VideoPCA (Stretcu and Leordeanu, 2015). This initial segmentation is later refined via graph-methods (Rother et al., 2004).

Co-segmentation/Co-localization. The co-segmentation methods have been leveraged to segment similar objects across a set of videos. Typical methods rely on discerning common objects via the similarity of different features including visual appearance, motion, and shape. Using initial guessing of objects/segments, graphs, such as CRF and graph cuts, are considered to model relationships between them (Chen et al., 2012; Fu et al., 2014; Tsai et al., 2016; Zhang et al., 2014). In (Chen et al., 2012), the authors use relative intra-video motion extracted from dense optical flow, and inter-video co-features based on Gaussian Mixture Models (GMM). The common object, *i.e.*, the foreground, in two videos is isolated from the background using a Markov Random Field (MRF). It is solved iteratively via graph cuts while accounting for a unary term based on the distance to GMMs, and a pairwise energy based on feature distance weighted by the relative motion distance between super-voxels. The authors in (Fu et al., 2014) pursue a similar path. The method starts by generating class-independent proposals (Endres and Hoiem, 2010). To identify the foreground object in each frame, various object characteristics are used while accounting for inter- and intra-video coherence of the foreground. A co-selection graph is formulated as a CRF exploiting unary energy from motion (optical flow), and pairwise energy over the foreground that combines visual (histogram) and shape similarities. In (Zhang et al., 2014), co-segmentation in videos is performed by sampling, tracking, and matching object proposals using graphs. It prunes away noisy segments in each video by selecting proposal tracklets that are spatially salient and temporally consistent. An iterative grouping process is used to gather objects with similar shapes and appearances (histogram) within and across videos. The final object regions obtained are used to initialize a segmentation process (Fulkerson et al., 2009). The authors in (Tsai et al., 2016) use a pre-trained FCN to generate object-like tracklets which are linked for each object category via a graph. To define the corresponding relation between tracklets, a sub-modular optimization is formulated based on the similarities of the tracklets via object appearance, shape, and motion. To discover prominent objects in each video, tracklets are ranked based on their mutual similarities.

Other methods aim to co-localize objects in images or videos using bounding boxes instead of segments. In (Joulin et al., 2014), a large number of proposed bounding boxes is generated by relying on objectness (Alexe et al., 2012) with the goal being to select a single box that is more likely to contain the common object across images/videos. This is achieved by solving a quadratic problem using the Frank-Wolfe algorithm (Frank and Wolfe, 1956). Following (Tang et al., 2014), box similarity and discriminability are used as constraints. Additionally, to ensure consistency between consecutive frames, multiple object properties including visual appearance, position, and size are used. (Jerripothula et al., 2016) leverage co-saliency as a prior to filter out noisy bounding box proposals. The co-saliency map is derived from inter- and intra-video commonness, and total motion saliency maps. Potential proposals are used for tracklet generation, and tracklets with a high confidence score and exhibiting good spatiotemporal consistency yield the final localization.

WSOL in still images. CAM-based methods have emerged as a dominant approach for the WSOL (Choe et al., 2020; Rony et al., 2023) on still images. Early works focused on designing variant spatial pooling layers (Durand et al., 2017; 2016; Lin et al., 2013; Oquab et al., 2015; Pinheiro and Collobert, 2015; Sun et al., 2016; Zhou et al., 2016; 2018). An extension to a multi-instance learning framework (MIL) has been considered (Ilse et al., 2018). However, CAMs tend to under-activate by highlighting only small and the most discriminative parts of an object (Choe et al., 2020; Rony et al., 2023), diminishing its localization performance. To overcome this, different strategies have been considered, including data augmentation over input image or deep features (Belharbi et al., 2022b; Choe and Shim, 2019; Li et al., 2018; Mai et al., 2020; Singh and Lee, 2017; Wei et al., 2017; Yun et al., 2019; Zhang et al., 2018b; Zhu et al., 2017), as well as architectural changes (Gao et al., 2021; Ki et al., 2020; Lee et al., 2019; Xue et al., 2019; Yang et al., 2020; Zhang et al., 2020c). Recently, learning via pseudo-labels shown some potential, despite its reliance on noisy labels (Belharbi et al., 2022c;a; Meethal et al., 2020; Murtaza et al., 2022a;b; Wei et al., 2021; Zhang et al., 2020a; 2018c). Most previous methods used only forward information in a CNN, with some models designed to leverage backward information as well. These include biologically inspired methods (Cao et al., 2015; Zhang et al., 2018a), and gradient (Chattopadhyay et al., 2018; Fu et al., 2020; Jiang et al., 2021; Selvaraju et al., 2017) or confidence score aggregation methods (Desai and Ramaswamy, 2020; Naidu et al., 2020; Naidu and Michael, 2020; Wang et al., 2020). While CAM-based methods are successful on still images, they still require adaptation to leverage the temporal information in videos as in (Belharbi et al., 2023). In (Belharbi et al., 2023), the authors considered improving the quality of pseudo-labels by aggregating CAMs extracted from a sequence of frames into a single CAM that covers more ROIs. However, because of object motion, the benefit of their method is quickly diminished after two frames.

3 Proposed Approach

Notation. We denote by $\mathbb{D} = \{(\boldsymbol{V}, y)_i\}_{i=1}^{N}$ a training set of videos, where $\boldsymbol{V} = \{\boldsymbol{X}_t\}_{t=1}^{T}$ is a video with T frames, $\boldsymbol{X}_t : \Omega \subset \mathbb{R}^2$ is the t-th frame, Ω is a discrete image domain. The global video tag, *i.e.*, class label, is denoted $y \in \{1, \cdots, K\}$, with K being the number of classes. It is assumed that the video label y is transferred to all frames within the video. Our model follows a U-Net style architec-

ture (Ronneberger et al., 2015) with skip connections (Fig.1). It is composed of two parts: an encoder g with parameters $\boldsymbol{\theta'}$ image classification and a decoder f with parameters $\boldsymbol{\theta}$ for localization.

Our classifier g is composed of a backbone encoder for extracting features, as well as a classification scoring head. The per-class classification probabilities are denoted $g(\boldsymbol{X}) \in [0,1]^K$ where $g(\boldsymbol{X})_k = \Pr(k|\boldsymbol{X})$. This module is trained to classify independent frames via standard cross-entropy, $\min_{\boldsymbol{\theta'}} -\log(\Pr(y|\boldsymbol{X}))$. Its weights, $\boldsymbol{\theta'}$, are then frozen and used to produce a CAM $\boldsymbol{C}_t$ for the frame $\boldsymbol{X}_t$ using the true label y (Choe et al., 2020). This makes $\boldsymbol{C}_t$ semantically consistent with the video tag. It is used later to generate pseudo-labels to train the decoder f.

The localizer f is a decoder that yields two full-resolution CAMs that are softmaxed, and denoted $\boldsymbol{S}_t = f(\boldsymbol{X}_t) \in [0,1]^{|\Omega| \times 2}$. $\boldsymbol{S}_t^0, \boldsymbol{S}_t^1$ refer to the background and foreground maps, respectively. Let $\boldsymbol{S}_t(p) \in [0,1]^2$ denotes a row of matrix $\boldsymbol{S}_t$, with index $p \in \Omega$ indicating a point within Ω. We denote by $\{\boldsymbol{X}\}_t^n = \{\boldsymbol{X}_{t-n-1}, \cdots, \boldsymbol{X}_{t-1}, \boldsymbol{X}_t\}$ a set of n consecutive frames starting at time t, and $\{\boldsymbol{S}\}_t^n$ it's corresponding CAMs at the decoder output, in the same order. As described below, we consider terms at two levels to train our decoder: per-frame and across-frames.

Per-frame priors. At the frame level, we leverage three terms to improve the CAM localization performance and alleviate their common issues in still images including bloby and unbalanced activations (Choe et al., 2020; Rony et al., 2023).

Learning via pseudo-labels (PL). It is commonly known that strong activations in a CAM are more likely to be a foreground, while low activations are assumed to be a background (Zhou et al., 2016). We leverage this information to generate pixel-wise pseudo-labels for foreground and background regions using the CAM $\boldsymbol{C}_t$ of the classifier g. Different from common practices where ROIs are generated once and kept fixed (Kolesnikov and Lampert, 2016), we stochastically sample our ROIs. This has been shown to be more effective and avoids overfitting (Belharbi et al., 2022a). Therefore, at each SGD step, we randomly sample a foreground pixel using a multinomial distribution over strong activations assuming that an object is local. We use a uniform distribution over low activations to sample background pixels assuming that background regions are evenly distributed in an image. The location of these two random pixels is encoded in Ω'_t. The partially pseudo-labeled mask for the sample $\boldsymbol{X}_t$ is denoted $\boldsymbol{Y}_t$, where $\boldsymbol{Y}_t(p) \in \{0,1\}^2$ with labels 0 for background, 1 for foreground, and locations with unknown labels are encoded as unknown. At this stage, since the pre-trained classifier g is frozen, its CAM $\boldsymbol{C}_t$ is fixed and does not change during the training of f, allowing more stable sampling. Leveraging $\boldsymbol{Y}_t$ is achieved using partial cross-entropy,

$$\boldsymbol{H}_p(\boldsymbol{Y}_t, \boldsymbol{S}_t) = \\ -(1-\boldsymbol{Y}_t(p))\ \log(1-\boldsymbol{S}_t^0(p)) - \boldsymbol{Y}_t(p)\ \log(\boldsymbol{S}_t^1(p))\ . \tag{1}$$

Local consistency (CRF). Standard CAMs are low resolution leading to a bloby effect where activations do not align with the object's boundaries, thus contributing to poor localization (Belharbi et al., 2022c; Choe et al., 2020). To avoid this, we use a CRF loss (Tang et al., 2018) to push the activations of $\boldsymbol{S}_t$ to be locally consistent in terms of color and proximity. For an image frame $\boldsymbol{X}_t$ and its maps $\boldsymbol{S}_t$, the CRF loss is formulated as,

$$\mathcal{R}(\boldsymbol{S}_t, \boldsymbol{X}_t) = \sum_{r \in \{0,1\}} {\boldsymbol{S}_t^r}^\top \boldsymbol{W}_t\ (\boldsymbol{1} - \boldsymbol{S}_t^r)\ , \tag{2}$$

where $\boldsymbol{W}_t$ is an affinity matrix in which $\boldsymbol{W}[i,j]$ captures the color similarity and proximity between pixels i, j in the image $\boldsymbol{X}_t$. We use a Gaussian kernel to capture the color and spatial similarities (Krähenbühl and Koltun, 2011). The kernel is implemented via the permutohedral lattice (Adams et al., 2010) for fast computation.

Absolute size constraint (ASC). Unbalanced activations are common in CAMs (Belharbi et al., 2022c; Choe et al., 2020; Rony et al., 2023). Often, strong activations cover only small and the most discriminative parts of an object allowing the background to dominate. Alternatively, large parts of an image are activated as foreground (Rony et al., 2023). To avoid both these scenarios, we employ a global constraint over the CAM, in which we push the size of both regions, *i.e.*, the foreground and background, to be as large as possible in a competitive way. To this end, we use an Absolute Size Constraint (ASC) (Belharbi et al., 2022b) over the maps $\boldsymbol{S}_t$. This is achieved without requiring any knowledge about the object size or a prior on which region is larger (Pathak et al., 2015). This generic prior is formulated as inequality constraints which are then solved via a standard log-barrier method (Boyd and Vandenberghe, 2004),

$$\sum \boldsymbol{S}_t^r \geq 0\ , \quad r \in \{0,1\}\ , \tag{3}$$

where $\psi(\boldsymbol{S}_t^0) = \sum \boldsymbol{S}_t^0, \psi(\boldsymbol{S}_t^1)$ represent the area, *i.e.*, size, of the background and foreground regions, respectively. Using log-barrier function, we set,

$$\mathcal{R}_s(\boldsymbol{S}_t) = \sum_{r \in \{0,1\}} -\frac{1}{z} \log(\psi(\boldsymbol{S}_t^r))\ , \tag{4}$$

where $z > 0$ is a weight that is increased periodically.

Multi-frame prior: color cue. Given a sequence of frames $\{\boldsymbol{X}\}_t^n$, we aim to consistently and simultaneously align their corresponding CAMs $\{\boldsymbol{S}\}_t^n$ in term of *color*. Thus, we perform a co-localization of objects with similar colors across frames. This is translated by *explicitly* constraining the CAMs

$\{\boldsymbol{S}\}_t^n$ to activate similarly over similar color pixels across all the frames. This assumes that an object appearing in a video sequence maintains similar color. However, unlike (Belharbi et al., 2023), we do not assume minimal displacement of objects. This gives our method more flexibility to localize an object independently from its location. Additionally, this joint learning opens an explicit communication tunnel between CAMs, allowing a transfer and aggregation of knowledge. This can help attenuate localization errors caused by noisy pseudo-labels. We perform this constraint by connecting each pixel at each frame to all pixels in every frame via the *color term* of the CRF loss (Tang et al., 2018). This creates a fully connected graph between all pixels, allowing explicit communication. In practice, this is achieved by *stitching* all frames[3] $\{\boldsymbol{X}\}_t^n$ to build a single large image. We refer to the composite image by $Cat(\{\boldsymbol{X}\}_t^n) = \bar{\boldsymbol{X}}$, where $Cat(\cdot)$ is a stitching function. Similarly, we denote $Cat(\{\boldsymbol{S}\}_t^n) = \bar{\boldsymbol{S}}$ as the corresponding stitched CAMs done in the same order as $Cat(\{\boldsymbol{X}\}_t^n)$. The loss is formulated as,

$$\mathcal{R}_c(\{\boldsymbol{S}\}_t^n, \{\boldsymbol{X}\}_t^n) = \sum_{r\in\{0,1\}} \bar{\boldsymbol{S}}^{r\top}\, \boldsymbol{W}\, (\mathbf{1} - \bar{\boldsymbol{S}}^r)\,, \quad (5)$$

where $\boldsymbol{W}$ is the *color* similarity matrix between pixels of the stitched image $\bar{\boldsymbol{X}}$. We refer to this term as CoLoc. Minimizing Eq.5 pushes the sequence of CAMs $\{\boldsymbol{S}\}_t^n$ to be consistent, with respect to color, across the frames sequence $\{\boldsymbol{X}\}_t^n$.

Total training loss. Our final loss combines per-frame and multi-frame losses to be optimized simultaneously. It is formulated as,

$$\min_{\boldsymbol{\theta}} \quad \sum_{p\in\Omega'_t} \boldsymbol{H}_p(\boldsymbol{Y}_t, \boldsymbol{S}_t) + \lambda\, \mathcal{R}(\boldsymbol{S}_t, \boldsymbol{X}_t) + \mathcal{R}_s(\boldsymbol{S}_t) + \frac{\lambda_c}{|[\mathcal{R}_c]|}\, \mathcal{R}_c(\{\boldsymbol{S}\}_t^n, \{\boldsymbol{X}\}_t^n)\,, \quad (6)$$

where λ and λ_c are positive weighting coefficients. All the terms are optimized simultaneously via SGD.

We note that the magnitude[4] of the term $\mathcal{R}_c$ increases with the number of frames n. Large magnitudes can easily overpower other terms in Eq.6, hindering learning. In practice, this makes tuning the hyper-parameter λ_c critical and challenging. To reduce this strong dependency on n and stabilize learning, we propose an *adaptive* weight λ_c that automatically *scales down* this term via its magnitude $|[\mathcal{R}_c]|$. This is inspired by the recently proposed adaptive weight decay (Ghiasi et al., 2022) which uses the weights' norm and their gradient for adaptation. In this context, the operation $[\cdot]$ indicates that this value is now a constant and does not depend on any optimization parameters, *i.e.*, $\boldsymbol{\theta}$. As a result, this term is always constant $\lambda_c \frac{\mathcal{R}_c}{|[\mathcal{R}_c]|} = \pm\lambda_c$. However, its derivative is non-zero $\partial\left(\lambda_c \frac{\mathcal{R}_c}{|[\mathcal{R}_c]|}\right)/\partial\boldsymbol{\theta} = \frac{\lambda_c}{|[\mathcal{R}_c]|}\frac{\partial\mathcal{R}_c}{\partial\boldsymbol{\theta}}$. In this adaptive setup, the value λ_c simply amplifies the scaled gradient. Its practical values are $\lambda_c > 1$, making it easier to tune compared to its non-adaptive version, which will be demonstrated empirically later. Most importantly, the adaptive version creates less dependency on n.

After being trained using Eq.6, our model is applied on single frames using a simple forward pass for inference. This allows for fast inference and parallel computation over a video. Using standard procedures (Belharbi et al., 2023; Choe et al., 2020), a localization bounding box is extracted from the foreground CAM of the decoder $\boldsymbol{S}_t^1$.

[3] Frames are stitched horizontally or vertically.

[4] Typical magnitude of $\mathcal{R}_c$ is $2\cdot10^9$ for $n=2$, and can grow to $2\cdot10^{11}$ for $n=18$. As a result, adequate values of the non-adapative λ_c should be below $2\cdot10^{-9}$.

4 Results and Discussion

4.1 Experimental Methodology

Datasets. To evaluate our method, experiments were conducted on standard unconstrained video datasets for WSVOL. For training, videos are labeled globally via a class-tag. Frame bounding boxes are provided to evaluate localization. We used two challenging public datasets from YouTube[5]: YouTube-Object v1.0 (`YTOv1` (Prest et al., 2012)) and YouTube-Object v2.2 (`YTOv2.2` (Kalogeiton et al., 2016)). In all our experiments, we followed the same protocol as described in (Belharbi et al., 2023; Kalogeiton et al., 2016; Prest et al., 2012).

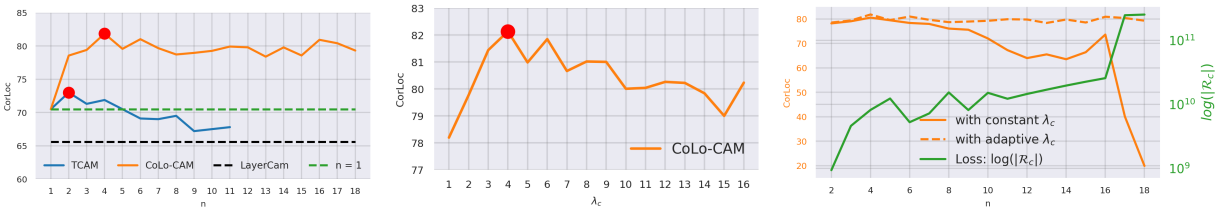
YouTube-Object v1.0 (`YTOv1`) (Prest et al., 2012): A dataset composed of videos collected from YouTube by querying the names of 10 classes. A class has between 9-24 videos with a duration ranging from 30 seconds to 3 minutes. It has 155 videos, each split into short-duration clips, *i.e.*, shots. There are 5507 shots with 571 089 frames in all. Each shot has only a few selected frames with bounding box annotations. The dataset has been divided by the authors into 27 test videos for a total of 396 bounding boxes, and 128 videos for training. Some videos from the trainset are considered for validation to perform early-stopping (Belharbi et al., 2023; Kalogeiton et al., 2016; Prest et al., 2012). We sampled 5 random videos per class to build a validation set with a total of 50 videos.

YouTube-Object v2.2 (`YTOv2.2`) (Kalogeiton et al., 2016): This is an extension of the `YTOv1` dataset with large and challenging test set. It is composed of more frames, 722 040 in total. The authors provided more bounding boxes annotation in this case. The dataset was divided by the authors into 106 videos for training, and 49 videos for test. Following (Belharbi et al., 2023), we sampled 3 random videos per class from the train to build a validation. The test set has more

[5] https://www.youtube.com

Dataset	Method *(venue)*	Aero	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Avg	Time/Frame
YTOv1	(Prest et al., 2012) *(cvpr)*	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5	N/A
	(Papazoglou and Ferrari, 2013) *(iccv)*	65.4	67.3	38.9	65.2	46.3	40.2	65.3	48.4	39.0	25.0	50.1	4s
	(Joulin et al., 2014) *(eccv)*	25.1	31.2	27.8	38.5	41.2	28.4	33.9	35.6	23.1	25.0	31.0	N/A
	(Kwak et al., 2015) *(iccv)*	56.5	66.4	58.0	76.8	39.9	69.3	50.4	56.3	53.0	31.0	55.7	N/A
	(Rochan et al., 2016) *(ivc)*	60.8	54.6	34.7	57.4	19.2	42.1	35.8	30.4	11.7	11.4	35.8	N/A
	(Tokmakov et al., 2016) *(eccv)*	71.5	74.0	44.8	72.3	52.0	46.4	71.9	54.6	45.9	32.1	56.6	N/A
	POD (Koh et al., 2016) *(cvpr)*	64.3	63.2	73.3	68.9	44.4	62.5	71.4	52.3	78.6	23.1	60.2	N/A
	(Tsai et al., 2016) *(eccv)*	66.1	59.8	63.1	72.5	54.0	64.9	66.2	50.6	39.3	42.5	57.9	N/A
	(Haller and Leordeanu, 2017) *(iccv)*	76.3	71.4	65.0	58.9	68.0	55.9	70.6	33.3	69.7	42.4	61.1	0.35s
	(Croitoru et al., 2019) (LowRes-Net$_{\text{iter1}}$) *(ijcv)*	77.0	67.5	77.2	68.4	54.5	68.3	72.0	56.7	44.1	34.9	62.1	0.02s
	(Croitoru et al., 2019) (LowRes-Net$_{\text{iter2}}$) *(ijcv)*	79.7	67.5	68.3	69.6	59.4	75.0	78.7	48.3	48.5	39.5	63.5	0.02s
	(Croitoru et al., 2019) (DilateU-Net$_{\text{iter2}}$) *(ijcv)*	85.1	72.7	76.2	68.4	59.4	76.7	77.3	46.7	48.5	46.5	65.8	0.02s
	(Croitoru et al., 2019) (MultiSelect-Net$_{\text{iter2}}$) *(ijcv)*	84.7	72.7	78.2	69.6	60.4	80.0	78.7	51.7	50.0	46.5	67.3	0.15s
	SPFTN (M) (Zhang et al., 2020b) *(tpami)*	66.4	73.8	63.3	83.4	54.5	58.9	61.3	45.4	55.5	30.1	59.3	N/A
	SPFTN (P) (Zhang et al., 2020b) *(tpami)*	**97.3**	27.8	81.1	65.1	56.6	72.5	59.5	**81.8**	79.4	22.1	64.3	N/A
	FPPVOS (Umer et al., 2021) *(optik)*	77.0	72.3	64.7	67.4	79.2	58.3	74.7	45.2	80.4	42.6	65.8	0.29s
	CAM (Zhou et al., 2016) *(cvpr)*	75.0	55.5	43.2	69.7	33.3	52.4	32.4	74.2	14.8	50.0	50.1	0.2ms
	GradCAM (Selvaraju et al., 2017) *(iccv)*	86.9	63.0	51.3	81.8	45.4	62.0	37.8	67.7	18.5	50.0	56.4	27.8ms
	GradCAM++ (Chattopadhyay et al., 2018) *(wacv)*	79.8	85.1	37.8	81.8	75.7	52.4	64.9	64.5	33.3	56.2	63.2	28.0ms
	Smooth-GradCAM++ (Omeiza et al., 2019) *(corr)*	78.6	59.2	56.7	60.6	42.4	61.9	56.7	64.5	40.7	50.0	57.1	136.2ms
	XGradCAM (Fu et al., 2020) *(bmvc)*	79.8	70.4	54.0	**87.8**	33.3	52.4	37.8	64.5	37.0	50.0	56.7	14.2ms
	LayerCAM (Jiang et al., 2021) *(ieee)*	85.7	**88.9**	45.9	78.8	75.5	61.9	64.9	64.5	33.3	56.2	65.6	17.9ms
	TCAM (Belharbi et al., 2023) *(wacv)*	90.5	70.4	62.2	75.7	**84.8**	**81.0**	81.0	64.5	70.4	50.0	73.0	18.5ms
	CoLo-CAM (ours)	90.4	74.0	**91.8**	**87.8**	78.7	80.9	**89.1**	74.1	**85.1**	**68.7**	**82.1**	18.5ms
YTOv2.2	(Haller and Leordeanu, 2017) *(iccv)*	76.3	68.5	54.5	50.4	59.8	42.4	53.5	30.0	53.5	**60.7**	54.9	0.35s
	(Croitoru et al., 2019) (LowRes-Net$_{\text{iter1}}$) *(ijcv)*	75.7	56.0	52.7	57.3	46.9	57.0	48.9	44.0	27.2	56.2	52.2	0.02s
	(Croitoru et al., 2019) (LowRes-Net$_{\text{iter2}}$) *(ijcv)*	78.1	51.8	49.0	60.5	44.8	62.3	52.9	48.9	30.6	54.6	53.4	0.02s
	(Croitoru et al., 2019) (DilateU-Net$_{\text{iter2}}$)*(ijcv)*	74.9	50.7	50.7	60.9	45.7	60.1	54.4	42.9	30.6	57.8	52.9	0.02s
	(Croitoru et al., 2019) (BasicU-Net$_{\text{iter2}}$)*(ijcv)*	82.2	51.8	51.5	62.0	50.9	64.8	55.5	45.7	35.3	55.9	55.6	0.02s
	(Croitoru et al., 2019) (MultiSelect-Net$_{\text{iter2}}$)*(ijcv)*	81.7	51.5	54.1	62.5	49.7	68.8	55.9	50.4	33.3	57.0	56.5	0.15s
	CAM (Zhou et al., 2016) *(cvpr)*	52.3	66.4	25.0	66.4	39.7	**87.8**	34.7	53.6	45.4	43.7	51.5	0.2ms
	GradCAM (Selvaraju et al., 2017) *(iccv)*	44.1	68.4	50.0	61.1	51.8	79.3	56.0	47.0	44.8	42.4	54.5	27.8ms
	GradCAM++ (Chattopadhyay et al., 2018) *(wacv)*	74.7	78.1	38.2	69.7	56.7	84.3	61.6	61.9	43.0	44.3	61.2	28.0ms
	Smooth-GradCAM++ (Omeiza et al., 2019) *(corr)*	74.1	83.2	38.2	64.2	49.6	82.1	57.3	52.0	51.1	42.4	59.5	136.2ms
	XGradCAM (Fu et al., 2020) *(bmvc)*	68.2	44.5	45.8	64.0	46.8	86.4	44.0	57.0	44.9	45.0	54.6	14.2ms
	LayerCAM (Jiang et al., 2021) *(ieee)*	80.0	84.5	47.2	**73.5**	55.3	83.6	71.3	60.8	55.7	48.1	66.0	17.9ms
	TCAM (Belharbi et al., 2023) *(wacv)*	79.4	**94.9**	75.7	61.7	68.8	87.1	75.0	62.4	72.1	45.0	72.2	18.5ms
	CoLo-CAM (ours)	**82.9**	92.2	**85.4**	67.7	**80.1**	85.7	**79.2**	**67.4**	**72.7**	58.2	**77.1**	18.5ms

Table 1: `CorLoc` localization accuracy on the `YTOv1` (Prest et al., 2012) and `YTOv2.2` (Kalogeiton et al., 2016) test sets.



(a) (b) (c)

Figure 2: Ablations on the `YTOv1` test set. **(a)**: Impact on `CorLoc` accuracy of the number of frames n. **(b)**: Impact on `CorLoc` accuracy of the *adaptive* λ_c. **(c)**: `CorLoc` accuracy on the `YTOv1` test set using constant and adaptive λ_c weight (*left y-axis*), and $\log(|\mathcal{R}_c|)$ (*right y-axis*). The *x-axis* is the number of frames n.

samples compared to YTOv1. It is composed of a total of 2 667 bounding box making it a much more challenging dataset.

Evaluation Metric. Localization accuracy was evaluated using the CorLoc metric (Belharbi et al., 2023; Deselaers et al., 2012). It describes the percentage of predicted bounding boxes that have an Intersection Over Union (IoU) between prediction and ground truth greater than half (IoU > 50%).

Implementation Details. We trained for 10 epochs with 32 mini-batch for all our experiments. We used ResNet50 (He et al., 2016) as a backbone. Images are resized to 256×256, and randomly cropped patches of size 224×224 for training. The temporal dependency n in Eq.5 is set via the validation set from $n \in \{2, \cdots, 18\}$. The hyper-parameter λ_c in Eq.6 is set from $\{1, \cdots, 16\}$ with the adaptive setup. In Eq.5, frames are stitched horizontally to build a large image. We set the CRF's hyper-parameter λ in Eq.6 to the same value as in (Tang et al., 2018) that is $2e^{-9}$. Its color and spatial kernel bandwidth are set to 15 and 100, respectively (Tang et al., 2018). We use the same color bandwidth for Eq.5. The initial value of the log-barrier coefficient, z in Eq.4, is set to 1, and then increased by a factor of 1.01 in each epoch with a maximum value of 10 as in (Belharbi et al., 2019; Kervadec et al., 2019). We selected the best learning rate from $\{0.1, 0.01, 0.001\}$. The classifier g was pre-trained on single frames.

Baseline Methods. For comparison, publicly available results were considered. We compared our proposed approach with different WSVOL methods (Croitoru et al., 2019; Haller and Leordeanu, 2017; Joulin et al., 2014; Kwak et al., 2015; Papazoglou and Ferrari, 2013; Prest et al., 2012; Rochan et al., 2016; Tokmakov et al., 2016; Tsai et al., 2016), POD (Koh et al., 2016), SPFTN (Zhang et al., 2020b), and FPPVOS (Umer et al., 2021). We also considered the performance of several CAM-based methods reported in (Belharbi et al., 2023): CAM (Zhou et al., 2016), GradCAM (Selvaraju et al., 2017), GradCam++ (Chattopadhyay et al., 2018), Smooth-GradCAM++ (Omeiza et al., 2019), XGradCAM (Fu et al., 2020), and LayerCAM (Jiang et al., 2021), which were all trained on single frames, *i.e.*, no temporal dependency was considered. LayerCAM method (Jiang et al., 2021) was employed to generate CAMs $\boldsymbol{C}_t$ used to create pseudo-labels, which were then used to build pseudo-labels $\boldsymbol{Y}_t$ (Eq.1). However, different CAM-based methods can be integrated with our approach.

4.2 Results

Comparison with the State-of-Art (Tab.1). Overall, our CoLo-CAM method outperformed other methods by a large margin, more so on YTOv1 compared to YTOv2.2. This highlights the difficulty of YTOv2.2. In terms of per-class performance, our method is competitive over most classes. In particular, we observe that our method achieved large improvement over the challenging class 'Train'. In general, our method and CAM methods are still behind on the 'Horse' and 'Aero' classes compared to the SPFTN method (Zhang et al., 2020b) which relies on optical flow to generate proposals. Both classes present different challenges. The former class shows with dense multi-instances, while the latter, *i.e.*, 'Aero', appears in complex scenes (airport), often with very large size, and occlusion. Our method has the same inference time as TCAM (Belharbi et al., 2023), and our new term in Eq.5 adds a small training time of ~ 81 ms per 64 frames (see supplementary material).

Ablation Studies.

Methods		CorLoc
Layer-CAM (Jiang et al., 2021) *(ieee)*		65.6
Single-frame	PL	68.5
	PL + CRF	69.6
	PL + ASC	66.2
	PL + ASC + CRF	70.5
Multi-frame	PL + ASC + CRF + CoLoc (Ours)	82.1
Improvement		+16.5

Table 2: Impact on CorLoc localization accuracy of different CoLo-CAM loss terms on the YTOv1 test set.

Impact of different loss terms (Tab.2). Without any spatiotemporal dependency, using pseud-labels (PL), CRF, and absolute size constraint (ASC) helped to improve localization performance. This brought up localization performance from 65.6% to 70.5%. However, adding our multi-frame term, *i.e.*, CoLoc, increased the performance up to 82.1% demonstrating its benefits.

Impact of n on localization performance (Fig.2a). We observe that both methods, TCAM (Belharbi et al., 2023) and ours, get better results when considering spatiotemporal information. However, TCAM reached its peak performance when using only $n = 2$ frames. Large performance degradation is observed when increasing n. This comes as a result of assuming that objects have minimal movement. On the opposite, our method improves when increasing n until it reaches its peak at $n = 4$. Different from TCAM, our method showed more robustness and stability with the increase of n since we only assume objects visual similarity.

Constant vs adaptive λ_c (Fig.2c). Note that in the constant setup, adequate λ_c can be determined using a blind search or given a prior on the magnitude loss. Both approaches are tedious especially when typical values of λ_c are below $2 * 10^{-9}$ creating a large search space. Exploring such space is computationally expensive. Using constant value held good results up to 6 frames, while, large n led to performance degradation since adequate values are required. Our adaptive scheme is more intuitive, and does not require a prior knowledge. It achieved constantly better and stable performance with less effort. Better results are obtained with λ_c values between 3, and 7 (Fig.2b). Performance degradation is observed when

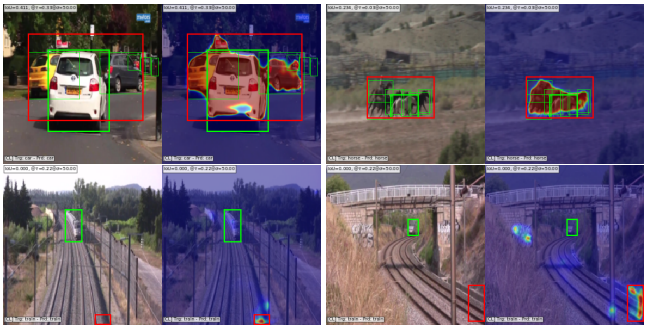


Figure 3: Typical challenges of our method. (Colors: Fig.4)

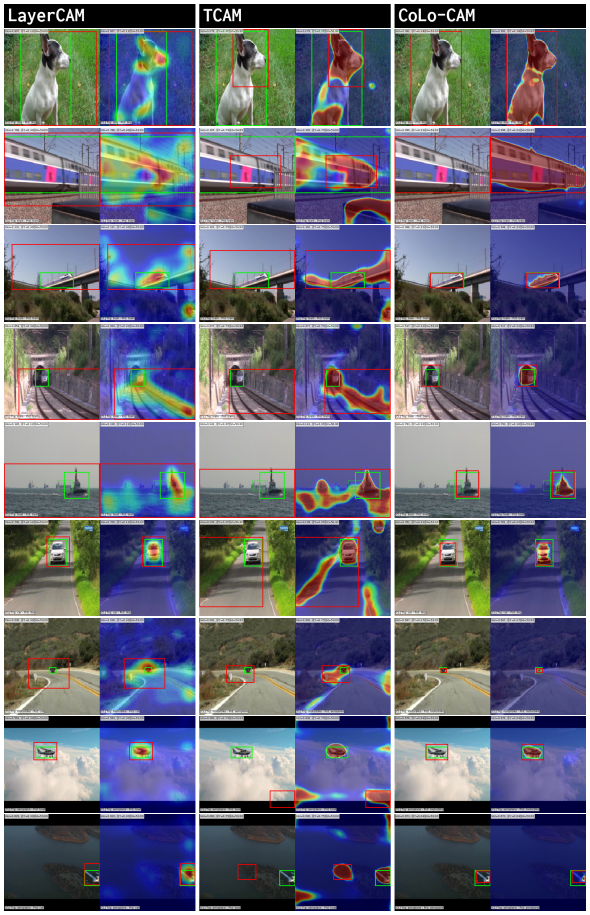


Figure 4: Localization examples of test sets frames. *Bounding boxes*: ground truth (green), prediction (red). The second column of each method is the predicted CAM over image.

using large amplification giving more importance to $\mathcal{R}_c$ term, allowing it to outweigh other terms which hindered training. In the opposite, using small amplification gave small push to this term, hence, less contribution.

Visual Results (Fig.4). Compared to TCAM (Belharbi et al., 2023) and LayerCAM (Jiang et al., 2021), we observe that our CAMs became more discriminative. This translates in several advantageous properties. Our CAMs showed sharp, more complete, and less noisy activations localized more precisely around objects. In addition, localizing small objects such as 'Bike', and large objects such as 'Train' becomes more accurate. CAM activations often suffer from co-occurrence issue (Belharbi et al., 2023; Rony et al., 2023) where consistently appearing contextual objects are confused with main objects. Our CAMs showed more robustness to this issue. This can be seen in the case of road vs 'Car'/'Bike', or water vs 'Boat'. Despite this improvement, our method still faces two main challenges (Fig.3). The first is the case of samples with dense and overlapped instances (top Fig.3). This often causes activations to spill across instances to form a large object hindering localization performance. The second issue concerns large localization errors (bottom Fig.3). Since our localization is mainly stimulated by discriminative cues from CAMs, it is still challenging to detect when a classifier points to the wrong object. This often leads to large errors that are difficult to correct in downstream use. Future works may consider leveraging classifier response over ROIs to ensure their quality. Additionally, object movement information can be combined with CAMs' cues.

5 Conclusion

We have proposed a new CAM-based approach for the WSVOL task. Using a color cue, we constrain the CAM response over a sequence of frames to be similar over similar pixels, assuming that an object maintains a similar color. This achieves explicit co-localization across frames. It is performed by minimizing a color-term of a CRF loss over a sequence of images/CAMs. In addition to our multi-frame constraint, we imposed per-frame local constraints to be then all optimized simultaneously. Empirical experiments showed the merits of our method and its robustness to long-term dependencies, leading to new state-of-the-art localization performance.

Acknowledgment

This research was supported in part by the Canadian Institutes of Health Research, the Natural Sciences and Engineering Research Council of Canada, and the Digital Research Alliance of Canada (alliancecan.ca).

The next supplementary materials are comprised of:

- detailed illustrations of our approach for training and inference (Sec.A);
- a discussion on pseudo-labels sampling (Sec.A);
- additional ablation studies (Sec.B.1); and
- additional visual results (Sec.B.2).

A Proposed Approach

Detailed illustrations for training and inference. Fig.5 presents our detailed model for training and inference. During the training, we consider $n = 3$ input frames. We use three per-frame terms that aim at improving localization at each frame, and without considering any spatiotemporal information. Using the classifier CAM $\boldsymbol{C}_t$, we sample pseudo-labels which are used at the pseudo-label term (PL). Additionally, we apply a CRF, and a size constraint (ASC). In parallel, we apply a multi-frame term ver all the frames allowing explicit communication between them. This is achieved by constraining the output CAMs of the frames to activate similarly over pixels with similar color.

At inference time, we simply consider single frames, *i.e.*, no spatiotemporal information. A bounding box is estimated from the foreground CAM, at the decoder level. In addition, frame classification scores are obtained from the encoder.

Learning via pseudo-labels (PLs). In this section, we provide more details on sampling pseudo-labels through the classifier's CAM $\boldsymbol{C}_t$. These labels are then used to train the decoder f, in combination with other terms. Through the discriminative property of the pseudo-labels, they initiate and stimulate object localization. In CAM-based methods for WSOL, it is common to assume that CAMs' strong activations are likely to point to the foreground, whereas low activations hint at background regions (Durand et al., 2017; Zhou et al., 2016).

In the following, we denote the foreground region as $\mathbb{C}_t^+$ which is computed by the operation $\mathcal{O}^+$. For simplicity and without adding new hyper-parameters, $\mathbb{C}_t^+$ is defined as the set of pixels with CAM activation $\boldsymbol{C}_t$ greater than the Otsu threshold (Otsu, 1979) estimated from $\boldsymbol{C}_t$. The background region, *i.e.*, $\mathbb{C}_t^-$, is the remaining set of pixels. Both regions are defined as follows,

$$\mathbb{C}_t^+ = \mathcal{O}^+(\boldsymbol{C}_t), \quad \mathbb{C}_t^- = \mathcal{O}^-(\boldsymbol{C}_t)\ . \tag{7}$$

Given the weak supervision, these regions are uncertain. For instance, the $\mathbb{C}_t^+$ region may hold only a small part of the foreground, along with the background parts. Consequently, the background region $\mathbb{C}_t^-$ may still have foreground parts. This uncertainty in partitioning the image foreground and background makes it unreliable for localization to directly fit (Kolesnikov and Lampert, 2016) the model to full regions. Instead, we pursue a stochastic sampling strategy over both regions to avoid overfitting and provide the model enough time for the emergence of consistent regions (Belharbi et al., 2022c;a). For each frame, and for an SGD step, we randomly sample a pixel from $\mathbb{C}_t^+$ as foreground, and a pixel from $\mathbb{C}_t^-$ as background to be pseudo-labels. We encode their location according to:

$$\Omega_t' = \mathcal{M}(\mathbb{C}_t^+) \cup \mathcal{U}(\mathbb{C}_t^-)\ , \tag{8}$$

where $\mathcal{M}(\mathbb{C}_t^+)$ is a multinomial sampling distribution over the foreground, and $\mathcal{U}(\mathbb{C}_t^-)$ is a uniform distribution over the background. This choice of distributions is based on the assumption that the foreground object is often concentrated in one place, while the background region is spread across the image. The partially pseudo-labeled mask for the sample $\boldsymbol{X}_t$ is referred to as Y_t, where $Y_t(p) \in \{0, 1\}^2$ with labels 0 for the background, and 1 for the foreground. Undefined regions are labeled as unknown. The pseudo-annotation mask Y_t is then leveraged to train the decoder using partial cross-entropy. Such stochastic pseudo-labels are expected to stimulate the learning of the filters and guide them to generalize and respond in the same way over regions with similar color/texture.

B Results and Discussion

B.1 Additional Ablation Studies.

Computation time of our multi-frame term (Eq.5) with respect to n (Fig.6). The computation of this loss term is divided on two types of devices (hybrid fashion): CPU and GPU. The left-side matrix product is performed on the GPU to accelerate the computation time, while the rest of the term is first computed on the CPU. Note that the transfer time from CPU to GPU is included in our complexity analysis. The overhead in processing time grows linearly with respect to the number of frames. However, this computational cost remains resealable for training in a realistic time, even with a large n. For example, $n = 64$ frames can be processed in ~ 81 ms.

Effect of frame sampling strategy (Tab.3). We explored different strategies to randomly sample n training frames from a video, with an emphasis on *frame diversity*:

- **Consecutive scheme**: A first frame is uniformly sampled from the video. Then, its previous $n - 1$ frames are also considered. This favors similar (less diverse) frames.
- **Interval scheme**: This is the opposite case of the consecutive scheme where the aim is to sample the most diverse frames. The video is split into n equal and disjoint intervals. Then, a single frame is uniformly sampled from each interval.
- **Gaussian scheme**: This middle-ground scheme lies between consecutive and interval strategies. We sample n random frames via a Gaussian distribution centered in the middle of a video.

All the sampling is done without repetition. In Tab.3, consecutive sampling performs the best, suggesting that our co-localization term is more beneficial when frames are similar. Note that using diverse frames via Gaussian or interval scheme still yielded good performance.

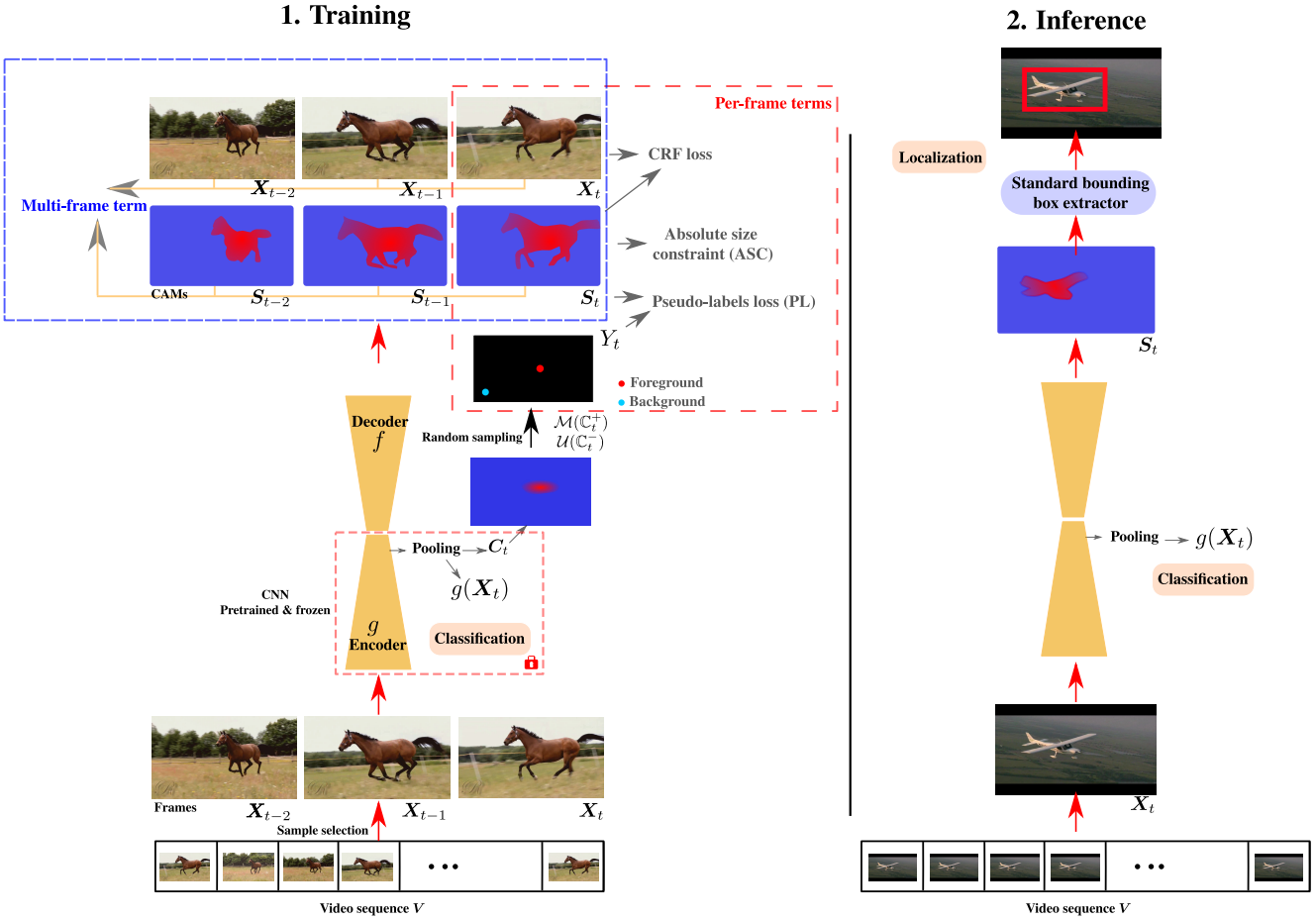


Figure 5: Training and inference with our proposed CoLo-CAM method. The single-frame and multi-frame training loss terms are illustrated with $n = 3$ frames. *Left: training phase*: It combines the per-frame terms composed of pseudo-labels loss (PL), absolute size constraint (ASC), and CRF loss, as well as the multi-frame term. *Right: inference phase*: It operates at a single frame with no temporal dependency, and predicts bounding box localization and classification. Notation and per-frame and multi-frame terms are described in Section 3.

Methods	CorLoc
Consecutive	82.1
Gaussian	80.2
Interval	79.8

Table 3: Impact on localization accuracy (CorLoc) of different frame sampling strategies on the YTOv1 test set.

B.2 Visual Results.

Fig.7, 8 displays more visual results. Images show that our method yielded sharp and less noisy CAMs. However, multi-instances and complex scenes remain a challenge.

In a separate file, we provide demonstrative videos which show the localization of objects. They highlight additional challenges not visible in still images, in particular:

- **Empty frames**. Some frames are without objects, yet still show class activations. This issue is the result of the weak global annotation of videos, and the assumption that all frames inherit that same label. It highlights the importance of detecting such frames and treating them accordingly, starting with classifier training. Discarding these frames will help the classifier better discern objects and reduce noisy CAMs. Without any additional supervision, WSVOL is still a difficult task. One possible solution is to leverage the per-class probabilities of the classifier to assess whether an object is present or not in a frame. Such likelihood can be exploited to discard frames or weight loss terms over them.
- **Temporal consistency**. Although our method brought a quantitative and qualitative improvement, there are some failure cases. In some instances,

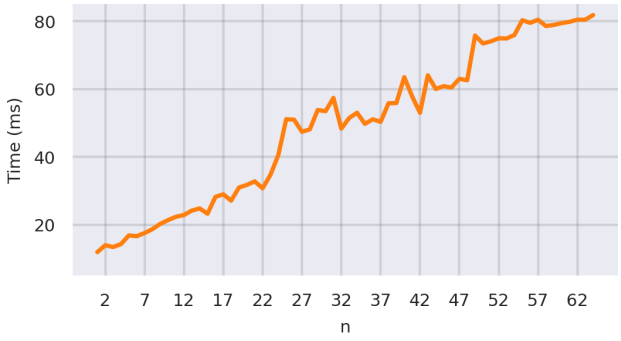


Figure 6: Computation time of our multi-frame loss term (Eq.5) in function of number of frames n. Computation devices: CPU (Intel(R) Xeon(R), 48 cores), and GPU (NVIDIA Tesla P100). Single image frame size: 224×224.

we observed inconsistencies between consecutive frames. This error is characterized by a large shift in localization where the bounding box (and the CAM activation), moves drastically. This often happens when objects become partially or fully occluded, the appearance of other instances, the scene becomes complex, or zooming in the image makes the object too large. Such setups lead to a *sudden shift* of the CAM's focus, leading to an abrupt change of localization which is undesirable. This drawback is mainly inherited from the inference procedure that is achieved at a single frame and without accounting for spatiotemporal information allowing such errors. Future works may consider improving the inference procedure by leveraging spatiotemporal information at the expense of inference time.

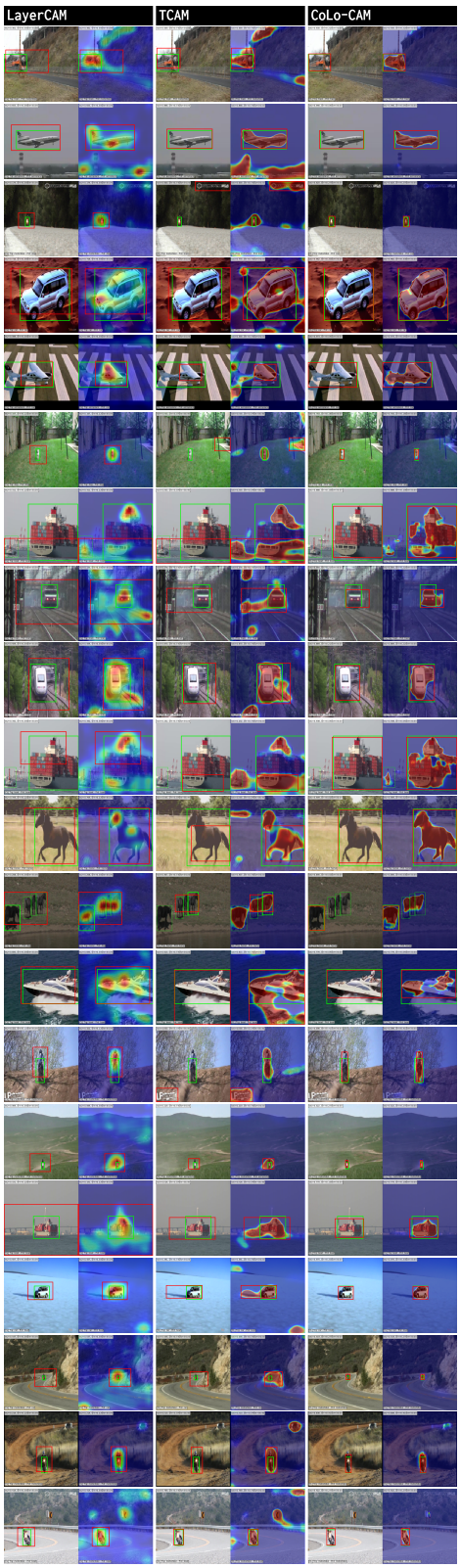


Figure 7: Additional localization examples of test sets frames (`YTOv1`, `YTOv2.2`). *Bounding boxes*: ground truth (green), prediction (red). The second column of each method is the predicted CAM over image.

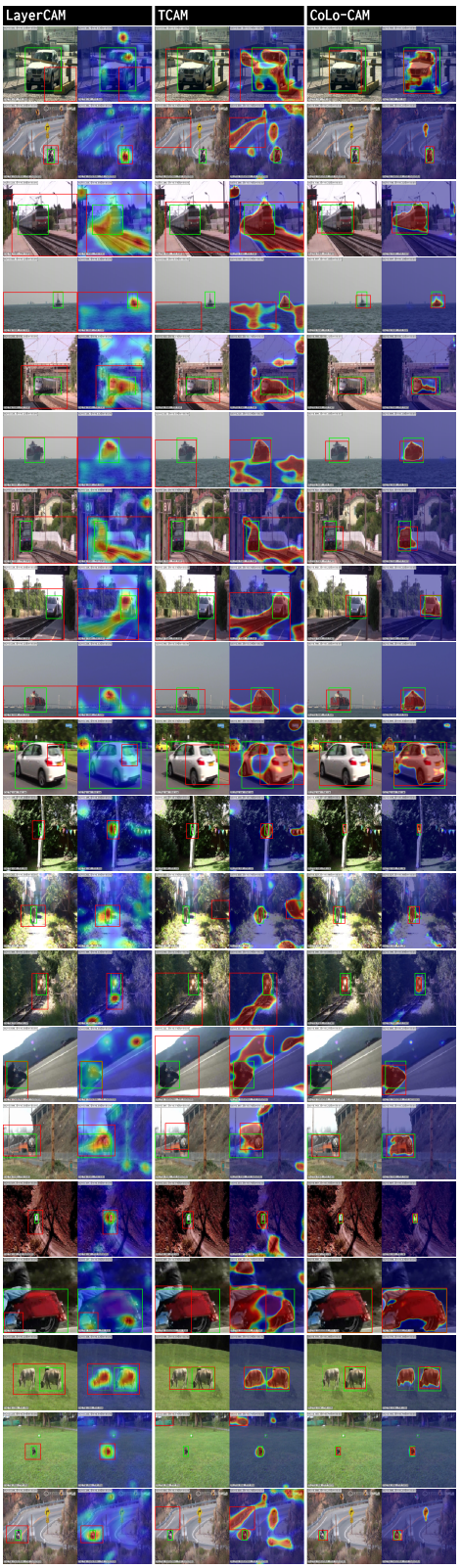


Figure 8: Additional localization examples of test sets frames (`YTOv1`, `YTOv2.2`). *Bounding boxes*: ground truth (green), prediction (red). The second column of each method is the predicted CAM over image.

References

Adams, A., Baek, J., and Davis, M. A. (2010). Fast high-dimensional filtering using the permutohedral lattice. *Comput. Graph. Forum*, 29(2):753–762.

Alexe, B., Deselaers, T., and Ferrari, V. (2012). Measuring the objectness of image windows. *TPAMI*, 34(11):2189–2202.

Banica, D., Agape, A., Ion, A., and Sminchisescu, C. (2013). Video object segmentation by salient segment chain composition. In *ICCVW*.

Belharbi, S., Ben Ayed, I., McCaffrey, L., and Granger, E. (2019). Deep ordinal classification with inequality constraints. *CoRR*, abs/1911.10720.

Belharbi, S., Ben Ayed, I., McCaffrey, L., and Granger, E. (2023). Tcam: Temporal class activation maps for object localization in weakly-labeled unconstrained videos. In *WACV*.

Belharbi, S., Pedersoli, M., Ben Ayed, I., McCaffrey, L., and Granger, E. (2022a). Negative evidence matters in interpretable histology image classification. In *MIDL*.

Belharbi, S., Rony, J., Dolz, J., Ben Ayed, I., McCaffrey, L., and Granger, E. (2022b). Deep interpretable classification and weakly-supervised segmentation of histology images via max-min uncertainty. *IEEE Transactions on Medical Imaging*, 41:702–714.

Belharbi, S., Sarraf, A., Pedersoli, M., Ben Ayed, I., McCaffrey, L., and Granger, E. (2022c). F-cam: Full resolution class activation maps via guided parametric upscaling. In *WACV*.

Bergmann, P., Meinhardt, T., and Leal-Taixé, L. (2019). Tracking without bells and whistles. In *ICCV*.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Brox, T. and Malik, J. (2010). Object segmentation by long term analysis of point trajectories. In *ECCV*.

Cao, C., Liu, X., Yang, Y., et al. (2015). Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks. In *ICCV*.

Chang, X., Yang, Y., Long, G., Zhang, C., and Hauptmann, A. G. (2016). Dynamic concept composition for zero-example event detection. In *AAAI*.

Chattopadhyay, A., Sarkar, A., Howlader, P., and Balasubramanian, V. N. (2018). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*.

Chen, D., Chen, H., and Chang, L. (2012). Video object cosegmentation. In *Multimedia Conference*.

Chen, Y., Cao, Y., Hu, H., and Wang, L. (2020). Memory enhanced global-local aggregation for video object detection. In *CVPR*.

Choe, J., Oh, S. J., Lee, S., Chun, S., Akata, Z., and Shim, H. (2020). Evaluating weakly supervised object localization methods right. In *CVPR*.

Choe, J. and Shim, H. (2019). Attention-based dropout layer for weakly supervised object localization. In *CVPR*.

Croitoru, I., Bogolin, S.-V., and Leordeanu, M. (2019). Unsupervised learning of foreground object segmentation. *International Journal of Computer Vision*, 127(9):1279–1302.

Desai, S. and Ramaswamy, H. (2020). Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *WACV*.

Deselaers, T., Alexe, B., and Ferrari, V. (2012). Weakly supervised localization and learning with generic knowledge. *IJCV*, 100(3):275–293.

Durand, T., Mordan, T., Thome, N., and Cord, M. (2017). Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *CVPR*.

Durand, T., Thome, N., and Cord, M. (2016). Weldon: Weakly supervised learning of deep convolutional neural networks. In *CVPR*.

Endres, I. and Hoiem, D. (2010). Category independent object proposals. In *ECCV*.

Felzenszwalb, P., Girshick, R., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645.

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110.

Fu, H., Xu, D., Zhang, B., and Lin, S. (2014). Object-based multiple foreground video co-segmentation. In *CVPR*.

Fu, R., Hu, Q., Dong, X., Guo, Y., Gao, Y., and Li, B. (2020). Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. In *BMVC*.

Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *ICCV*.

Gao, W., Wan, F., Pan, X., Peng, Z., Tian, Q., Han, Z., Zhou, B., and Ye, Q. (2021). TS-CAM: token semantic coupled attention map for weakly supervised object localization. In *ICCV*.

Ghiasi, A., Shafahi, A., and Ardekani, R. (2022). Adaptive weight decay: On the fly weight decay tuning for improving robustness. *CoRR*, abs/2210.00094.

Haller, E. and Leordeanu, M. (2017). Unsupervised object segmentation in video by efficient selection of highly probable positive features. In *ICCV*.

Han, M., Wang, Y., Chang, X., and Qiao, Y. (2020). Mining inter-video proposal relations for video object detection. In *ECCV*.

Hartmann, G., Grundmann, M., Hoffman, J., Tsai, D., Kwatra, V., Madani, O., Vijayanarasimhan, S., Essa, I. A., Rehg, J. M., and Sukthankar, R. (2012). Weakly supervised learning of object segmentations from web-scale video. In *ECCV*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

Ilse, M., Tomczak, J., and Welling, M. (2018). Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR.

Jerripothula, K. R., Cai, J., and Yuan, J. (2016). CATS: co-saliency activated tracklet selection for video co-localization. In *ECCV*.

Jiang, P., Zhang, C., Hou, Q., Cheng, M., and Wei, Y. (2021). Layercam: Exploring hierarchical class activation maps for localization. *IEEE Trans. Image Process.*, 30:5875–5888.

Joulin, A., Tang, K., and Fei-Fei, L. (2014). Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*.

Kalogeiton, V., Ferrari, V., and Schmid, C. (2016). Analysing domain shift factors between videos and images for object detection. *TPAMI*, 38(11):2327–2334.

Kervadec, H., Dolz, J., Yuan, J., Desrosiers, C., Granger, E., and Ben Ayed, I. (2019). Constrained deep networks: Lagrangian optimization via log-barrier extensions. *CoRR*, abs/1904.04205.

Ki, M., Uh, Y., Lee, W., and Byun, H. (2020). In-sample contrastive learning and consistent attention for weakly supervised object localization. In *ACCV*.

Koh, Y. J., Jang, W., and Kim, C. (2016). POD: discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In *CVPR*.

Kolesnikov, A. and Lampert, C. H. (2016). Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*.

Krähenbühl, P. and Koltun, V. (2011). Efficient inference in fully connected crfs with gaussian edge potentials. In J.S.-Taylor, Zemel, R., Bartlett, P., Pereira, F. N., and Weinberger, K., editors, *NeurIPS*.

Kwak, S., Cho, M., Laptev, I., Ponce, J., and Schmid, C. (2015). Unsupervised object discovery and tracking in video collections. In *ICCV*.

Lee, J., Kim, E., Lee, S., Lee, J., and Yoon, S. (2019). Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *CVPR*.

Lee, Y., Kim, J., and Grauman, K. (2011). Key-segments for video object segmentation. In *ICCV*.

Li, K., Wu, Z., Peng, K., et al. (2018). Tell me where to look: Guided attention inference network. In *CVPR*.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *coRR*, abs/1312.4400.

Liu, X., Tao, D., Song, M., Ruan, Y., Chen, C., and Bu, J. (2014). Weakly supervised multiclass video segmentation. In *CVPR*.

Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., and Kim, T. (2021). Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448.

Mai, J., Yang, M., and Luo, W. (2020). Erasing integrated learning: A simple yet effective approach for weakly supervised object localization. In *CVPR*.

Manen, S., Guillaumin, M., and Gool, L. V. (2013). Prime object proposals with randomized prim's algorithm. In *ICCV*.

Meethal, A., Pedersoli, M., Belharbi, S., and Granger, E. (2020). Convolutional stn for weakly supervised object localization and beyond. In *ICPR*.

Murtaza, S., Belharbi, S., Pedersoli, M., Sarraf, A., and Granger, E. (2022a). Constrained sampling for class-agnostic weakly supervised object localization. In *Montreal AI symposium*.

Murtaza, S., Belharbi, S., Pedersoli, M., Sarraf, A., and Granger, E. (2022b). Discriminative sampling of proposals in self-supervised transformers for weakly supervised object localization. *CoRR*, abs/2209.09209.

Naidu, R., Ghosh, A., Maurya, Y., K, S. R. N., and Kundu, S. S. (2020). IS-CAM: integrated score-cam for axiomatic-based explanations. *CoRR*, abs/2010.03023.

Naidu, R. and Michael, J. (2020). SS-CAM: smoothed score-cam for sharper visual feature localization. *CoRR*, abs/2006.14255.

Omeiza, D., Speakman, S., Cintas, C., and Weldemariam, K. (2019). Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *CoRR*, abs/1908.01224.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2015). Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Papazoglou, A. and Ferrari, V. (2013). Fast object segmentation in unconstrained video. In *ICCV*.

Pathak, D., Krahenbuhl, P., and Darrell, T. (2015). Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*.

Pinheiro, P. and Collobert, R. (2015). From image-level to pixel-level labeling with convolutional networks. In *CVPR*.

Prest, A., Leistner, C., Civera, J., Schmid, C., and Ferrari, V. (2012). Learning object class detectors from weakly annotated video. In *CVPR*.

Rochan, M., Rahman, S., Bruce, N. D. B., and Wang, Y. (2016). Weakly supervised object localization and segmentation in videos. *Image and Vision Computing*, 56:1–12.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*.

Rony, J., Belharbi, S., Dolz, J., Ben Ayed, I., McCaffrey, L., and Granger, E. (2023). Deep weakly-supervised learning methods for classification and localization in histology images: A survey. *Machine Learning for Biomedical Imaging*, 2:96–150.

Rother, C., Kolmogorov, V., and Blake, A. (2004). "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*.

Shao, F., Chen, L., Shao, J., Ji, W., Xiao, S., Ye, L., Zhuang, Y., and Xiao, J. (2022). Deep learning for weakly-supervised object detection and localization: A survey. *Neurocomputing*, 496:192–207.

Singh, K. and Lee, Y. (2017). Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*.

Stretcu, O. and Leordeanu, M. (2015). Multiple frames matching for object discovery in video. In *BMVC*.

Sun, C., Paluri, M., Collobert, R., Nevatia, R., and Bourdev, L. (2016). Pronet: Learning to propose object-specific boxes for cascaded neural networks. In *CVPR*.

Sundaram, N., Brox, T., and Keutzer, K. (2010). Dense point trajectories by gpu-accelerated large displacement optical flow. In *ECCV*.

Tang, K., Sukthankar, R., Yagnik, J., and Fei-Fei, L. (2013). Discriminative segment annotation in weakly labeled video. In *CVPR*.

Tang, K. D., Joulin, A., Li, L., and Fei-Fei, L. (2014). Co-localization in real-world images. In *CVPR*.

Tang, M., Perazzi, F., Djelouah, A., Ben Ayed, I., Schroers, C., and Boykov, Y. (2018). On regularized losses for weakly-supervised cnn segmentation. In *ECCV*.

Tokmakov, P., Alahari, K., and Schmid, C. (2016). Weakly-supervised semantic segmentation using motion cues. In *ECCV*.

Tsai, Y.-H., Zhong, G., and Yang, M.-H. (2016). Semantic co-segmentation in videos. In *ECCV*.

Umer, S., Dawood, H., Yousaf, M. H., Dawood, H., and Ahmad, H. (2021). Efficient foreground object segmentation from video by probability weighted moments. *Optik*, 229:166251.

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., and Hu, X. (2020). Score-cam: Score-weighted visual explanations for convolutional neural networks. In *CVPR workshop*.

Wei, J., Wang, Q., Li, Z., Wang, S., Zhou, S. K., and Cui, S. (2021). Shallow feature matters for weakly supervised object localization. In *CVPR*.

Wei, Y., Feng, J., Liang, X., Cheng, M.-M., Zhao, Y., and Yan, S. (2017). Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*.

Xu, C., Xiong, C., and Corso, J. J. (2012). Streaming hierarchical video segmentation. In *ECCV*.

Xue, F., Tan, Z., Zhu, Y., Ma, Z., and Guo, G. (2022). Coarse-to-fine cascaded networks with smooth predicting for video facial expression recognition. In *CVPRW*.

Xue, H., Liu, C., Wan, F., Jiao, J., Ji, X., and Ye, Q. (2019). Danet: Divergent activation for weakly supervised object localization. In *ICCV*.

Yan, Y., Xu, C., Cai, D., and Corso, J. J. (2017). Weakly supervised actor-action segmentation via robust multi-task ranking. In *CVPR*.

Yang, S., Kim, Y., Kim, Y., and Kim, C. (2020). Combinational class activation maps for weakly supervised object localization. In *WACV*.

Yun, S., Han, D., Chun, S., Oh, S., Yoo, Y., and Choe, J. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*.

Zhang, C., Cao, Y., and Wu, J. (2020a). Rethinking the route towards weakly supervised object localization. In *CVPR*.

Zhang, D., Han, J., Jiang, L., Ye, S., and Chang, X. (2017). Revealing event saliency in unconstrained video collection. *IEEE Trans. Image Process.*, 26(4):1746–1758.

Zhang, D., Han, J., Yang, L., and Xu, D. (2020b). Spftn: A joint learning framework for localizing and segmenting objects in weakly labeled videos. *TPAMI*, 42(2):475–489.

Zhang, D., Javed, O., and Shah, M. (2014). Video object co-segmentation by regulated maximum weight cliques. In *ECCV*.

Zhang, J., Bargal, S. A., Lin, Z., et al. (2018a). Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102.

Zhang, W., Qiu, F., Wang, S., Zeng, H., Zhang, Z., An, R., Ma, B., and Ding, Y. (2022). Transformer-based multimodal information fusion for facial expression analysis. In *CVPRW*.

Zhang, X., Wei, Y., Feng, J., Yang, Y., and Huang, T. (2018b). Adversarial complementary learning for weakly supervised object localization. In *CVPR*.

Zhang, X., Wei, Y., Kang, G., Yang, Y., and Huang, T. (2018c). Self-produced guidance for weakly-supervised object localization. In *ECCV*.

Zhang, X., Wei, Y., and Yang, Y. (2020c). Inter-image communication for weakly supervised localization. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J., editors, *ECCV*, Lecture Notes in Computer Science.

Zhang, Y., Chen, X., Li, J., Wang, C., and Xia, C. (2015). Semantic object segmentation via detection in weakly labeled video. In *CVPR*.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *CVPR*.

Zhou, Y., Zhu, Y., Ye, Q., Qiu, Q., and Jiao, J. (2018). Weakly supervised instance segmentation using class peak response. In *CVPR*.

Zhu, Y., Zhou, Y., Ye, Q., et al. (2017). Soft proposal networks for weakly supervised object localization. In *ICCV*.