# TCAM: Temporal Class Activation Maps for Object Localization in Weakly-Labeled Unconstrained Videos

Soufiane Belharbi[1], Ismail Ben Ayed[1], Luke McCaffrey[2], and Eric Granger[1]
[1] LIVIA, Dept. of Systems Engineering, École de technologie supérieure, Montreal, Canada
[2] Goodman Cancer Research Centre, Dept. of Oncology, McGill University, Montreal, Canada
soufiane.belharbi.1@ens.etsmtl.ca, {ismail.benayed, eric.granger}@etsmtl.ca,
luke.mccaffrey@mcgill.ca

## Abstract

*Weakly supervised video object localization (WSVOL) allows locating object in videos using only global video tags such as object classes. State-of-art methods rely on multiple independent stages, where initial spatio-temporal proposals are generated using visual and motion cues, and then prominent objects are identified and refined. The localization involves solving an optimization problem over one or more videos, and video tags are typically used for video clustering. This process requires a model per video or per class making for costly inference. Moreover, localized regions are not necessary discriminant because these methods rely on unsupervised motion methods like optical flow, or discarded video tags from optimization. In this paper, we leverage the successful class activation mapping (CAM) methods, designed for WSOL based on still images. A new Temporal CAM (TCAM) method is introduced for training a discriminant deep learning (DL) model to exploit spatio-temporal information in videos, using an CAM-Temporal Max Pooling (CAM-TMP) aggregation mechanism over consecutive CAMs. In particular, activations of regions of interest (ROIs) are collected from CAMs produced by a pretrained CNN classifier, and generate pixel-wise pseudo-labels for training a decoder. In addition, a global unsupervised size constraint, and local constraint such as CRF are used to yield more accurate CAMs. Inference over single independent frames allows parallel processing of a clip of frames, and real-time localization. Extensive experiments[1] on two challenging YouTube-Objects datasets with unconstrained videos indicate that CAM methods (trained on independent frames) can yield decent localization accuracy. Our proposed TCAM method achieves a new state-of-art in WSVOL accuracy, and visual results suggest that it can be adapted for subsequent tasks, such as object detection and tracking.*

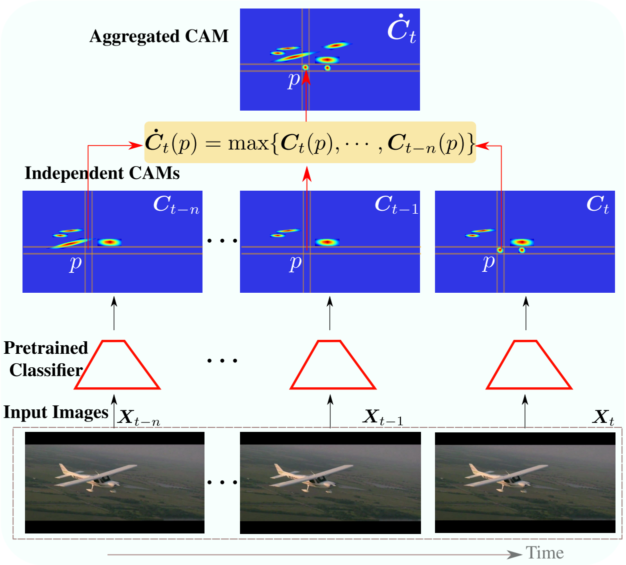

Figure 1: Example of CAM-Temporal Max Pooling (CAM-TMP) module for ROI aggregation of $n+1$ consecutive CAMs generated by a pretrained CNN classifier. It relies on the *maximum* activation at location $p$ across the independent CAMs to produce the output CAM, $\dot{C}_t$, that covers more discriminative parts. Notation is described in Sec.3.

## 1. Introduction

A massive amount of videos can be easily accessed on the internet thanks to the rapid growth of video sharing platforms [63, 69]. Therefore, the need to develop automatic methods to process and analyze these videos is of a great interest. The video object localization task plays a critical role toward video content understanding. It can improve the performance of subsequent tasks such as video summarization [83], event detection [11], video object detection [14, 26, 63], and visual object tracking [8, 44].

Videos are often captured in the wild with varying quality,

[1] Code: https://github.com/sbelharbi/tcam-wsol-video.

and mostly without constraints (moving objects and camera, viewpoint changes, decoding artifacts, and editing effects). However, while being abundant, exploiting these videos for down-stream tasks is still an ongoing challenge, mainly due to the high cost of annotation. Compared to still images, labeling videos represents a more difficult and expensive process, as videos often contain a large number of frames. For the object localization task, bounding boxes are required for each frame. Given this cost, videos are typically weakly-labeled [30, 72] using class tags. A weak label is defined for the entire video, and often describes the main object or concept appearing in the video, without detailed spatio-temporal information. However, this translates to noisy/corrupted labels at frame level – labels are assigned to an entire video, while only some of its frames may contain the object of interest. Although using weak labels drastically reduce the cost of annotation, it creates additional challenges for visual recognition tasks like object localization.

Weakly-supervised learning has emerged as an important paradigm to leverage coarse or global annotations like video tags, mitigating the need for bounding boxes annotation. Despite the importance of WSVOL, it has seen limited research [32, 37, 38, 57, 58, 84]. Instead, the literature on weakly supervised video object segmentation (WSVOS) has dominated [17, 22, 25, 43, 72, 71, 73, 79, 85], where bounding boxes are assumed to be produced via post-processing.

Most of the existing WSVOL methods are conventional, except for [17, 71, 84]. They usually generate spatio-temporal segments or proposals using visual and motion cues, and then prominent objects are identified and refined through post-processing. While they typically yield good performance, these methods present several limitations. They all require multiple sequential stages, and are not trained in an end-to-end manner. They are also costly at inference time since solutions are often optimized over a single video, or a cluster of videos from the same class. Moreover, they require building one model per class or per video, which is cumbersome in real-world applications, and scales poorly to a large number of classes. These methods are often non-discriminative – video labels are not *explicitly* used in a differentiable way to extract objects, and instead only to cluster videos of same class. Consequentially, localized objects are not necessarily aligned *semantically* with the video tag. Similarly, since almost all methods use motion cues, such as optical flow [40, 68], they are prone to this alignment issue since such motion cues do not account for semantics. Lastly, motion information in unconstrained videos is very noisy due to movement of camera and objects.

To alleviate the aforementioned limitations, a new Temporal CAM (TCAM) method is proposed to train a single discriminative DL model through weakly-supervised learning. Our method requires only video tag annotations, and does not rely on additional assumptions. It is motivated by the success of Class Activation Mapping (CAM) methods [91], applied for weakly-supervised object localization (WSOL) tasks on still images [7, 16, 20, 24, 31, 64, 75, 91]. Using only global image-class labels, CAM methods allow training a DL model end-to-end, in a differentiable way, to localize discriminative image regions. Consequentially, localized ROIs are aligned with the semantic label of the image. At inference time, a CNN can rapidly classify an image and localized the corresponding ROI. This method scales well to a large number of classes, making it suitable for real-world applications. However, these methods are limited to single images, and cannot exploit the temporal dependency between frames in videos. To leverage this spatio-temporal information, a new CAM Temporal Max-Pooling (CAM-TMP) mechanism is introduced to aggregate the ROIs from a sequence of CAMs (see Fig.1). Our CAM-TMP simulates *union* operation over ROIs in each frame by gathering ROIs from consecutive CAM, and thereby providing a better coverage of an object.

Our TCAM method relies on a U-Net style architecture [59] to classify an image, and localize the corresponding ROI through full-resolution CAMs for better accuracy (see Fig.2). Using a pre-trained CNN to classify frames, our DL model is trained over a *sequence* of successive frames, where CAM-TMP is used to accumulate ROIs. These are employed to generate reliable pseudo-labels for training the decoding architecture at the pixel-level. Following common practice [20, 91], strong activations in a CAM are considered as foreground, while low activations are background. At each Stochastic Gradient Descent (SGD) step, we randomly sample foreground (FG) and background (BG) pixel pseudo-labels within independent CAMs [7, 5] to train the decoder. Such random sampling allows exploring FG/BG regions, and promotes the emergence of consistent CAMs. In contrast to standard CAMs for WSOL, our CAM-TMP generate activation maps that provide a better coverage of the true objects, which leads to better sampling of FG and BG pixels. To mitigate common issues of CAMs such as small ROI, we use unsupervised size prior [7, 5, 6, 55] as a global constraints to encourage growth of both FG and BG regions, and avoid learning unbalanced CAMs. CRF loss [70] is also used to align CAMs with object boundaries by leveraging statistical properties of the image such as pixel color and proximity between pixels. Once the DL model is trained, inference is performed rapidly over independent frames, without considering temporal dependencies. This is more suitable for real-time applications than other state-of-art WSVOL methods since TCAM is not required to process an entire video to localize within a single frame.

Our work aims to improve the state-of-art performance in WSVOL, while encouraging new research in this area. **Our main contributions are summarized as follows.**

**(1)** We introduce the TCAM method, the first CAM-based method for WSVOL. In contrast with state-of-art WSVOL methods, TCAM allows training a single discriminative DL (U-Net style) model to process all classes at once. Our method is trained over unconstrained videos, each one annotated with a global class tag, and without any additional assumptions. Once trained, TCAM is able to rapidly predict the bounding box location for an object estimated based on any CAM method, along with the corresponding class label on each independent frame.

**(2)** Unlike standard CAMs which are limited to still images, TCAM leverages spatio-temporal information in a sequence of CAMs. Using CAM-TMP module, we extract relevant ROIs from a sequence of generic CAMs provided by a pretrained CNN classifier. CAM-TMP then yields a single accurate CAM with better coverage of an object. Our loss exploits this CAM for training the decoder, by randomly sampling from its FG/BF regions. Additional constraint losses, including size prior and CRF, are used to obtained balanced and accurate CAMs. Note that our TCAM is generic, and can be integrated on top of any CAM method.

**(3)** Extensive experiments conducted on two challenging public datasets – YouTube-Objects v1.0 and v2.2 – that are comprised of unconstrained videos indicate that: (a) standard CAM methods designed for WSOL on still images can achieve a high level of localization accuracy on frames from test set videos; (b) our TCAM method can achieves state-of-art in WSVOL accuracy. Results suggest that TCAM can be adapted for challenging downstream tasks, such as visual object detection and tracking.

## 2. Related Work

**Weakly Supervised Video Object Localization.** The limited amount of research in this category are often based on non-discriminative and non-deep models. These methods [37, 38, 57, 58, 84] initialize and select prominent proposals to be refined while considering spatio-temporal consistency constraints using mainly visual appearance and motion features. Different methods use prominent proposals as supervision to train a localizer [57, 84]. Others rely on segmentation [58] followed by additional refinement using GrabCut [61]. Single or cluster of videos are considered at once for optimization. For instance, POD method [37] considers an iterative approach to localize a primary object in a video assumed to appear in most frames but not in all frames. It is achieved while empty frames are identified. Region proposals are initially generated via [2]. Each proposal is bisected into foreground and background. Models for foreground, background, and primary object are built. An iterative scheme is setup to refine each model in an evolutionary manner. The final primary model is used to select candidate proposals and locate the bounding box. Recently, SPFTN [84] considers a deep learning (DL) framework that jointly learns to segment and localize objects with noisy supervision estimation using advanced optical flow [40]. Self-paced learning is considered to alleviate the ambiguity/noisy supervision. Other methods [32] rely on co-localization of common object over a set of videos or images.

**Weakly Supervised Video Object Segmentation.** Most methods are non-deep and non-discriminative based models. They undergo multiple steps to perform segmentation. They often operate on a single video or a cluster of videos, and bounding boxes are obtained via post-processing.

A set of methods initiate the learning by extracting independent spatio-temporal segments [27, 69, 77, 79] using for instance unsupervised methods [3, 77] or generate proposals [90] using pretrained detectors [90]. These object-parts are then gathered using different features which mainly include visual appearance and motion cues, while preserving temporal consistency. This is often done using graph-based models such as Conditional random field (CRF) or GrabCut-like approach [61]. DL models are rarely used. For instance, authors in [43] propose a nearest neighbor-based label transfer between videos to deal with multi-class video segmentation. Videos are first segmented into spatio-temporal supervoxels [77] which then represented in high dimensional feature space using color, texture, and motion. A multi-video graph is built, and using appearance, this graph model encourages label smoothness between spatio-temporal adjacent supervoxels in the same video and supervoxels with similar appearance across other videos. This yields a final pixel segmentation. M-CNN [71] combines motion cues with a fully convolutional network (FCN). A Gaussian mixture model is used to estimate foreground appearance potentials via motion [54]. These potentials are combined with the FCN prediction to estimate label predictions of foreground using a GrabCut-like approach [61]. These labels are used to fit the FCN. Authors use a fine-tuning stage over only few videos.

Other methods leverage co-segmentation to segment an object based on its occurrence on multiple images. A dominant approach is to use inter- and intra videos visual and motions cues to find common segments. Graphs, such as CRF and graph cuts, are used to model relations between variant segments [13, 22, 72, 85]. For example, authors in [72] generate object-like tracklets using a pretrained FCN. After collecting tracklets from all videos, they are linked for each object category via a graph. A sub-modular optimization is formulated to define the corresponding relation between tracklets based on their similarities while accounting for object properties such as appearance, shape, and motion. After maximizing this sub-modular function, tracklets are ranked using their mutual similarities allowing discovery of prominent objects in each video.

While all previous methods use labels to cluster videos of the same class, other methods do not use labels. However, the general process is roughly the same since previous methods do not exploit labels explicitly in their optimization. An initial guess of foreground regions is estimated [17, 25, 54, 73]. This is achieved either using motion cues via [68] such as in [54], Principal Component Analysis (PCA) such in [25, 73], or using VideoPCA algorithm [66] as in [17]. This initial guess is not necessarily discriminative. A final segmentation is then obtained by refinement using graphs [61]. For instance, authors in [17] propose a DL model. It is based on an iterative learning process where at each iteration, a CNN teacher is trained to discover object in videos. Object discovery is achieved using VideoPCA algorithm [66] which leverages spatio-temporal consistency in videos using appearance, shape, movement, and location of objects. Estimated foreground by the the teacher are fed to a CNN student for supervised training. Through iteration, several students are built and replace object discovery providing more reliable object segmentation.

**Weakly Supervised Object Localization in Still Images.** Early work in WSOL [60] focused on designing different spatial pooling layers, including Global Average Pooling (GAP) [42], weighed GAP [91], max-pooling [52], LSE [56, 67], PRM [92], WILDCAT [20, 21], and multi-instance learning pooling (MIL) [29]. However, these methods attained their limitation because CAMs can cover only small discriminative parts of the object. Subsequent work aimed to improve this aspect by refining the CAMs. This achieved through three different ways: *(1) via data augmentation* by perturbating input image such as in HaS [64], CutMix [81], AE [76], ACoL [87], MEIL [45], and MaxMin [6]; or by perturbating features as in SPN [93], GAIN [41], and ADL [16], or *(2) via architectural changes* such as in NL-CCAM [80], FickleNet [39], DANet [78], I$^2$C [89], ICL [35], and TS-CAM [24], or *(3) by using pseudo-labels for fine-tuning* such as in SPG [88], PSOL [82], SPOL [75], FCAM [7], NEGEV [5], and DiPS [48, 47]. Other methods aim to produce the bounding box directly without CAMs [46]. All the aforementioned methods extract localization from forward pass in the model. Other methods rely on forward and backward pass to estimate CAMs. This includes methods that *(1) are biologically inspired* such as feedback layer [10], and Excitation-backprop [86], or *(2) rely on gradient-aggregation* such as GradCAM [62], GradCam++ [12], XGradCAM [23], and LayerCAM [31], or *(3) use confident-aggregation* to avoid gradient saturation [1, 36] such as Ablation-CAM [18], Score-CAM [74], SS-CAM [50], and IS-CAM [49]. Despite the success of these methods, they are limited to still images and they are not equipped to leverage temporal information in videos.

Our proposal benefits from the simplicity of CAM methods, which provide single discriminative DL model for the WSOL task, to mitigate several issues in WSVOL. In addition, our TCAM method leverages the spatio-temporal information in videos.

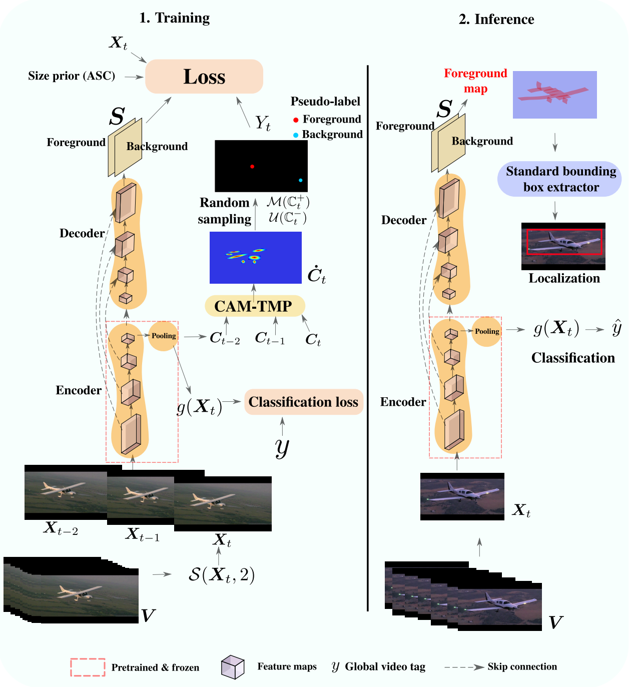

Figure 2: Our proposed TCAM method. *Left*: training with temporal dependency $n = 2$. *Right*: inference (no temporal dependency). See notation in Sec.3.

## 3. Proposed Approach

**Notation.** Let $\mathbb{D} = \{(\boldsymbol{V}, y)_i\}_{i=1}^{N}$ denotes a training set of videos, where $\boldsymbol{V} = \{\boldsymbol{X}_t\}_{t=1}^{t=T}$ is an input video with $T$ frames, and $\boldsymbol{X}_t : \Omega \subset \mathbb{R}^2$ is the $t$th frame in the video; $y \in \{1, \cdots, K\}$ is the video global class label, with $K$ the number of classes, and $\Omega$ is a discrete image domain. Assume that all frames inherit the same class label as the video global class tag. Our model is a U-Net style architecture [59] (Fig.2). It is composed of two parts: (a) classification module $g$ with parameters $\boldsymbol{W}$. It performs image classification. (b) segmentation module (decoder) $f$ with parameters $\boldsymbol{\theta}$. It outputs two CAMs, one for the foreground and the other for background. The classifier $g$ is composed of an encoder backbone for building features, and a pooling head to yield classification scores. We denote by $g(\boldsymbol{X}) \in [0,1]^K$ the per-class classification probabilities where $g(\boldsymbol{X})_k = \Pr(k|\boldsymbol{X})$. The classifier $g$ is trained using standard cross-entropy to correctly classify independent frames, $\min_{\boldsymbol{W}} \; -\log(\boldsymbol{g}(\boldsymbol{X})[y])$.

Once trained, its weights $\boldsymbol{W}$ are frozen, and not considered for future training. Classifier $g$ can yield a CAM of the target $y$, referred to as $\boldsymbol{C}$. We note $\boldsymbol{C}_t$ as the corresponding CAM of the the frame $\boldsymbol{X}_t$ at time $t$. The decoder generates softmax activation maps denoted as $\boldsymbol{S}_t = f(\boldsymbol{X}_t) \in [0,1]^{|\Omega| \times 2}$. Note that $\boldsymbol{S}_t^0, \boldsymbol{S}_t^1$ 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 $\Omega$. The operation $\mathcal{S}(\cdot, n)$ provides the set of $n$ previous neighbors of an element in the same video, plus the element itself. For instance, $\mathcal{S}(\boldsymbol{X}_t, n) = \{\boldsymbol{X}_t, \boldsymbol{X}_{t-1}, \cdots, \boldsymbol{X}_{t-n}\}$ is the set of $n$ previous frames of the frame $\boldsymbol{X}_t$, and $\mathcal{S}(\boldsymbol{C}_t, n) = \{\boldsymbol{C}_t, \boldsymbol{C}_{t-1}, \cdots, \boldsymbol{C}_{t-n}\}$ is the set of $n$ previous CAMs of the CAM $\boldsymbol{C}_t$.

**CAM Temporal Max-Pooling (CAM-TMP).** A sequence of frames in a video often captures the same scene with minimal variations. Therefore, objects within the scene have small displacement. However, this small change can cause CAMs to vary slightly, and highlight different minimal parts of the object as ROI. We leverage this behavior to build a single CAM, $\dot{\boldsymbol{C}}_t$, that covers more parts at once. This CAM will be used later for sampling foreground/background regions. To this end, we propose an aggregation method between consecutive CAMs where we perform a *union* operation between all spotted ROI in each CAM. This is achieved by taking the *maximum* CAM activation through time over activation in the same position of a sequence of CAMs (Fig.1). This is similar to spatial max-pooling operation, commonly used in CNNs, that seeks the presence of the object in small spatial neighborhood. Our temporal max-pooling seeks to determine whether one of the CAMs has activated over the object in a sequence of CAMs. At spatial position $p$, we formulate our CAM-TMP by taking the maximum across all CAMs at the same position,

$$\dot{\boldsymbol{C}}_t(p) = \max\{\boldsymbol{C}^1(p), \cdots, \boldsymbol{C}^{n+1}(p)\}, \; \boldsymbol{C}^i \in \mathcal{S}(\boldsymbol{C}_t, n), \tag{1}$$

where $\boldsymbol{C}^i$ is the $i$th element of the set $\mathcal{S}(\boldsymbol{C}_t, n)$, and $p \in \Omega$.

**Sampling Pseudo-Labels.** To guide the training of the decoder $f$, we exploit pixel-wise pseudo-supervision collected from the previously built CAM $\dot{\boldsymbol{C}}_t$ for the corresponding frame $\boldsymbol{X}_t$. We rely on the common assumption that strong activations in a CAM are more likely to be foreground, and low activations are considered background [7, 5, 20, 91]. We denote $\mathbb{C}_t^+$ as foreground region, estimated via the operation $\mathcal{O}^+$. It is determined as pixels with activations greater than Otsu threshold [53] estimated over $\dot{\boldsymbol{C}}_t$. The leftover region, estimated via the operation $\mathcal{O}^-$, is considered more likely background $\mathbb{C}_t^-$.

$$\mathbb{C}_t^+ = \mathcal{O}^+(\dot{\boldsymbol{C}}_t), \quad \mathbb{C}_t^- = \mathcal{O}^-(\dot{\boldsymbol{C}}_t) \; . \tag{2}$$

Both foreground and background regions are noisy and uncertain. The region $\mathbb{C}_t^-$ is still likely to contain part of the object. Similarly, $\mathbb{C}_t^+$ may still contain background. Due to this uncertainty, we avoid to directly fit these regions to the model. Instead, we consider a stochastic sampling over each region to avoid overfitting and allow the emergence of consistent regions [7, 5]. For each frame, and at each SGD step, we randomly select one pixel as foreground pseudo-label, and another single pixel as background pseudo-label. Their location is represented in,

$$\Omega_t' = \mathcal{M}(\mathbb{C}_t^+) \, \cup \, \mathcal{U}(\mathbb{C}_t^-) \; , \tag{3}$$

where $\mathcal{M}(\mathbb{C}_t^+)$ is a multinomial sampling distribution function over foreground region that samples a single location using the magnitude of pixels activations located exclusively in $\mathbb{C}_t^+$. Therefore, strong activation are more likely to be sampled as foreground.

Uniform sampling distribution $\mathcal{U}(\mathbb{C}_t^-)$ is used to sample a single background pixel from $\mathbb{C}_t^-$. We favor uniform random exploration of background region since the background is evenly distributed over the image. However, foreground region is only distributed over one place where the object is located. We denote by $Y_t$ the *partially* pseudo-labeled mask for the sample $\boldsymbol{X}_t$, where $Y_t(p) \in \{0,1\}^2$ with labels 0 for background, and 1 for foreground. This mask holds the sampled locations in Eq. 3 and their pseudo-labels. Locations with unknown pseudo-labels are encoded as unknown.

**Overall Training Loss.** Our training loss considers a frame $\boldsymbol{X}_t$ and its $n$ previous frames, *i.e.* $\mathcal{S}(\boldsymbol{X}_t, n)$, to leverage spatio-temporal information in a video. The time position $t$ is uniformly and randomly sampled in the video. The loss is composed of three parts. **a)** pixel-wise alignment using pseudo-label $Y_t$. This is achieved using partial cross-entropy,

$$\begin{aligned} &\boldsymbol{H}_p(Y_t, \boldsymbol{S}_t) = \\ &-(1 - Y_t(p)) \; \log(1 - \boldsymbol{S}_t^0(p)) - Y_t(p) \; \log(\boldsymbol{S}_t^1(p)) \; . \end{aligned} \tag{4}$$

**b)** To avoid the common unbalanced problem CAMs $\boldsymbol{S}_t$, where the background dominates the foreground (or the opposite), a global constraint is considered – the absolute size constraint (ASC) [6] over both regions. We do not assume whether the background is larger than the foreground [55] nor the opposite. This constraint pushes both regions to be large, and it is formulated as inequality constraints which are then solved via a standard log-barrier method [9]. **c)** To avoid trivial solution of ASC, where half the image is foreground and the other half is background, we use an additional local term that leverage pixels statistics including color and proximity. In particular, the CRF loss [70], denoted by $\mathcal{R}$ is included to ensure that CAM activations are consistent with the object boundaries and sampling regions.

Our overall total loss is formulated as,

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \sum_{p \in \Omega_t'} \boldsymbol{H}_p(Y_t, \boldsymbol{S}_t) + \lambda \, \mathcal{R}(\boldsymbol{S}_t, \boldsymbol{X}_t) \; , \\ \text{s.t.} \quad & \sum \boldsymbol{S}_t^r \geq 0 \; , \quad r \in \{0,1\} \; , \end{aligned} \tag{5}$$

where $\sum \boldsymbol{S}_t^0, \sum \boldsymbol{S}_t^1$ are the area size of the background and foreground regions, respectively.

The training of our method (Eq. 5) only requires the video global tag $y$ to train the classifier $g$, and properly estimate the pseudo-label mask $Y_t$ corresponding to the correct object class labeled in the video. This ensures that the semantic meaning of the foreground in $\boldsymbol{S}_t^1$ is aligned with the true label $y$. The spatio-temporal dependency between CAMs is leveraged in Eq. 1 to compute $\dot{\boldsymbol{C}}_t$ which is then used to sample $Y_t$. Our final trained model is evaluated on single independent frames, thereby producing a class prediction for the object in the fame, along with its spatial localization $\boldsymbol{S}_t^1$. Therefore, frames can be processed in parallel saving more inference time. Standard methods may be used for bounding box estimation in CAMs (Fig.2, *right*) [15].

## 4. Results and Discussion

### 4.1. Experimental Methodology

**Datasets.** For evaluation, we perform experiments on unconstrained video datasets for WSVOL task where videos are labeled globally using class label for training, and frame bounding boxes are provided for evaluation. In particular, we consider two public challenging datasets: YouTube-Object v1.0 (`YTOv1` [57]) and v2.2 (`YTOv2.2` [33] ) datasets. We follow common protocol for WSVOL task [33, 57].

*YouTube-Object v1.0 (`YTOv1`) [57]:* This dataset is composed of videos collected from YouTube[2] by querying for the names of 10 object classes. Each class has between 9 and 24 videos with a duration that varies from 30 seconds to 3 minutes. It contains 155 videos where each video is split into short duration clips named shots. There are 5507 shots, with each gathering multiple frames, reaching in total 571 089 frames. In each shot, only few frames are annotated with a bounding box to localized object of interest. The authors divided the dataset into 27 testing videos with a total of 396 labeled bounding boxes, and 128 video for training. It is common to use part of the training videos as validation set. In our experiments, we consider 5 random videos per class which amounts to a total of 50 videos for validation.

*YouTube-Object v2.2 (`YTOv2.2`) [33]:* This is an extension and improvement of `YTOv1`. It contains more frames, 722 040 frames in total. More importantly, authors provided more bounding boxes annotations. They divided the dataset into 106 videos for training, and 49 videos for test. For validation set, we consider, in our case, 3 random videos per class from the trainset. Compared to `YTOv1`, the test set contains much more annotation. It holds 1 781 frames with bounding boxes annotation, and a total of 2 667 bounding boxes. This makes this release much more challenging.

**Evaluation Measure.** For localization performance, `CorLoc` metric [19] is used. It represents the percentage of predicted bounding boxes that have an Intersection Over Union (IoU) between prediction and ground truth greater than half (IoU > 50%). In addition, standard classification accuracy, `CL`, is used to measure classification performance. It is measured over frames with bounding boxes.

**Implementations Details.** In all our experiments, we train for 100 epochs with 32 mini-batch size. Following WSOL task [15], we used ResNet50 [28] as a backbone. Images are resized to $256 \times 256$, then randomly cropped to $224 \times 224$ for training. The temporal dependency $n$ in Eq.1 is set via the validation set from the set $n \in \{1, \cdots, 10\}$. In Eq.5, the hyper-parameter $\lambda$ for the CRF is set to the same value as in [70] that is $2e^{-9}$. For log-barrier optimization, hyper-parameter $t$ is set to the same value as in [4, 34]. It is initialized to 1, and increased by a factor of 1.01 in each epoch with a maximum value of 10. In all experiments, we used a learning rate in $\{0.1, 0.01, 0.001\}$ using SGD for optimization. Our classifier is pretrained on independent frames. In all our experiments, and due to the large number of redundant frames per video, we randomly select a different frame in each shot at each epoch. This allows training CNNs over videos in a reasonable time.

**Baseline Methods.** For validation, we compare our method to available results. In particular, we compare to [17, 25, 32, 38, 54, 57, 58, 71, 72], POD [37], SPFTN [84], and FPPVOS [73]. Additionally, we implemented several CAM-based methods for further comparison. This includes CAM [91], GradCAM [62], GradCam++ [12], Smooth-GradCAM++ [51], XGradCAM [23], and LayerCAM [31]. CAM-based methods are trained on independent frames. In all our experiments, we use LayerCAM [31] to generate CAMs used to build a complete CAM $\dot{\boldsymbol{C}}_t(p)$ (Eq.1), which is then used to build pseudo-labels $Y_t$ (Eq.5). Note that our method is generic. It can be used with any CAM method.

### 4.2. Results

**Comparison with State-of-Art**[3]**.** Tab. 1 presents the results obtained on both datasets `YTOv1`, and `YTOv2.2`. We first note that CAM-based methods are very competitive compare to previous state-of-the-art methods. In particular, GradCAM++ [12] and LayerCAM [31] achieved an average localization performance of `CorLoc` of $63.1\%, 65.6\%$ over `YTOv1`, and $61.2\%, 66.0\%$ over `YTOv2.2`, respectively. Previous state-of-the-art methods yielded $67.3\%$, and $56.5\%$, respectively. This demonstrates the benefit of discriminative training of CAMs even though they are not aware of temporal dependency. Training our CAM-based method with temporal awareness between frames has boosted the localization performance furthermore reaching new state-of-the-art results. The same table shows as well that all methods suffer a discrepancy in performance between different objects where some classes are easier than others. For instance, the class 'train' seems very difficult. In CAM-based methods, we noticed that 'train' localization is often mistaken

[2] `https://www.youtube.com`

[3] The supplementary material provides some additional results, demonstrative videos.

| Dataset | Method *(venue)* | Aero | Bird | Boat | Car | Cat | Cow | Dog | Horse | Mbike | Train | Avg | Time/Frame |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| `YTOv1` | [57] *(cvpr,2012)* | 51.7 | 17.5 | 34.4 | 34.7 | 22.3 | 17.9 | 13.5 | 26.7 | 41.2 | 25.0 | 28.5 | N/A |
| | [54] *(iccv,2013)* | 65.4 | 67.3 | 38.9 | 65.2 | 46.3 | 40.2 | 65.3 | 48.4 | 39.0 | 25.0 | 50.1 | 4s |
| | [32] *(eccv,2014)* | 25.1 | 31.2 | 27.8 | 38.5 | 41.2 | 28.4 | 33.9 | 35.6 | 23.1 | 25.0 | 31.0 | N/A |
| | [38] *(iccv,2015)* | 56.5 | 66.4 | 58.0 | 76.8 | 39.9 | 69.3 | 50.4 | 56.3 | 53.0 | 31.0 | 55.7 | N/A |
| | [58] *(ivc,2016)* | 60.8 | 54.6 | 34.7 | 57.4 | 19.2 | 42.1 | 35.8 | 30.4 | 11.7 | 11.4 | 35.8 | N/A |
| | [71] *(eccv,2016)* | 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 [37] *(cvpr,2016)* | 64.3 | 63.2 | 73.3 | 68.9 | 44.4 | 62.5 | 71.4 | 52.3 | 78.6 | 23.1 | 60.2 | N/A |
| | [72] *(eccv,2016)* | 66.1 | 59.8 | 63.1 | 72.5 | 54.0 | 64.9 | 66.2 | 50.6 | 39.3 | 42.5 | 57.9 | N/A |
| | [25] *(iccv,2017)* | 76.3 | 71.4 | 65.0 | 58.9 | 68.0 | 55.9 | 70.6 | 33.3 | 69.7 | 42.4 | 61.1 | 0.35s |
| | [17] (LowRes-Net$_{\text{iter1}}$) *(ijcv,2019)* | 77.0 | 67.5 | 77.2 | 68.4 | 54.5 | 68.3 | 72.0 | 56.7 | 44.1 | 34.9 | 62.1 | 0.02s |
| | [17] (LowRes-Net$_{\text{iter2}}$) *(ijcv,2019)* | 79.7 | 67.5 | 68.3 | 69.6 | 59.4 | 75.0 | 78.7 | 48.3 | 48.5 | 39.5 | 63.5 | 0.02s |
| | [17] (DilateU-Net$_{\text{iter2}}$) *(ijcv,2019)* | 85.1 | 72.7 | 76.2 | 68.4 | 59.4 | 76.7 | 77.3 | 46.7 | 48.5 | 46.5 | 65.8 | 0.02s |
| | [17] (MultiSelect-Net$_{\text{iter2}}$) *(ijcv,2019)* | 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) [84] *(tpami,2020)* | 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) [84] *(tpami,2020)* | **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 [73] *(optik,2021)* | 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 [91] *(cvpr,2016)* | 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 [62] *(iccv,2017)* | 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++ [12] *(wacv,2018)* | 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++ [51] *(corr,2019)* | 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 [23] *(bmvc,2020)* | 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 [31] *(ieee,2021)* | 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 (ours) | 90.5 | 70.4 | 62.2 | 75.7 | **84.8** | **81.0** | **81.0** | 64.5 | 70.4 | 50.0 | **73.0** | 18.5ms |
| `YTOv2.2` | [25] *(iccv,2017)* | 76.3 | 68.5 | 54.5 | 50.4 | 59.8 | 42.4 | 53.5 | 30.0 | 53.5 | **60.7** | 54.9 | 0.35s |
| | [17] (LowRes-Net$_{\text{iter1}}$) *(ijcv,2019)* | 75.7 | 56.0 | 52.7 | 57.3 | 46.9 | 57.0 | 48.9 | 44.0 | 27.2 | 56.2 | 52.2 | 0.02s |
| | [17] (LowRes-Net$_{\text{iter2}}$) *(ijcv,2019)* | 78.1 | 51.8 | 49.0 | 60.5 | 44.8 | 62.3 | 52.9 | 48.9 | 30.6 | 54.6 | 53.4 | 0.02s |
| | [17] (DilateU-Net$_{\text{iter2}}$)*(ijcv,2019)* | 74.9 | 50.7 | 50.7 | 60.9 | 45.7 | 60.1 | 54.4 | 42.9 | 30.6 | 57.8 | 52.9 | 0.02s |
| | [17] (BasicU-Net$_{\text{iter2}}$)*(ijcv,2019)* | **82.2** | 51.8 | 51.5 | 62.0 | 50.9 | 64.8 | 55.5 | 45.7 | 35.3 | 55.9 | 55.6 | 0.02s |
| | [17] (MultiSelect-Net$_{\text{iter2}}$)*(ijcv,2019)* | 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 [91] *(cvpr,2016)* | 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 [62] *(iccv,2017)* | 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++ [12] *(wacv,2018)* | 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++ [51] *(corr,2019)* | 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 [23] *(bmvc,2020)* | 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 [31] *(ieee,2021)* | 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 (ours) | 79.4 | **94.9** | **75.7** | 61.7 | **68.8** | 87.1 | **75.0** | **62.4** | **72.1** | 45.0 | **72.2** | 18.5ms |

Table 1: Localization performance (`CorLoc`) on the `YTOv1` [57] and `YTOv2.2` [33] test sets.

with railway track since they often co-occur together. In addition, this object is often filmed from close range, at stations, leading to a large object, that often covers the entire frame making its localization difficult.

**Ablation Studies.** We performed an ablation study for key components of our loss function, using LayerCAM [31] as baseline to generate CAMs for pseudo-labels (see Tab.2). We observe that using only pseudo-labels improved the localization accuracy from $65.6\%$ to $68.5\%$. Adding CRF helps localization, but using only size constraint did not provide much benefits compared to the baseline alone. Combining pseudo-labels, CRF, and size constraint yielded the best localization performance of $70.5\%$ but without considering temporal dependency. Adding our temporal module, CAM-TMP, improves the localization accuracy up to $73\%$, indicating its benefit.

In addition, the impact of time range dependency is investigated (see Fig.3). As expected, considering previous frames ($n > 0$) helped in improving localization compared to looking only to instant frames ($n = 0$). However, long range dependency hampers the performance after $n = 1$. After $n = 4$, localization performance drops below the case of $n = 0$. This is thought to be caused by *object displacement*. Spatial locations in nearby frames cover typically the same objects. Therefore, ROI in CAMs are expected to land on same objects. Consequently, collecting ROI via Eq.1 is expected to be beneficial. However, moving to far away frames makes the same spatial location cover *different* objects, therefore collecting the wrong object. As a result, while our proposed module, *i.e*. CAM-TMP, can leverage temporal dependency in videos to improve localization, it is limited to short range frames. Nonetheless, using long range time dependency still yields better performance than the baseline method LayerCAM [31] (Fig.3). Based on our results, we recommend using short range dependency. We mention that such factor is strongly tied to the video frame rate. Using long range dependency in fast frame rate could be safe. However, slow frame rate should be considered with caution. We note that the information of video frame rate is not provided in the studied datasets.

| **Methods** | | `CorLoc` |
|---|---|---|
| Layer-CAM [31] *(ieee,2021)* | | 65.6 |
| $n = 0$ | Ours + $\mathbb{C}_t^+$ + $\mathbb{C}_t^-$ | 68.5 |
| | Ours + $\mathbb{C}_t^+$ + $\mathbb{C}_t^-$ + CRF | 69.6 |
| | Ours + $\mathbb{C}_t^+$ + $\mathbb{C}_t^-$ + ASC | 66.2 |
| | Ours + $\mathbb{C}_t^+$ + $\mathbb{C}_t^-$ + CRF + ASC | 70.5 |
| $n > 0$ | Ours + $\mathbb{C}_t^+$ + $\mathbb{C}_t^-$ + CRF + ASC + CAM-TMP | 73.0 |
| Improvement | | +7.4 |

Table 2: Localization accuracy (`CorLoc`) of TCAM with different losses on the `YTOv1` test set.

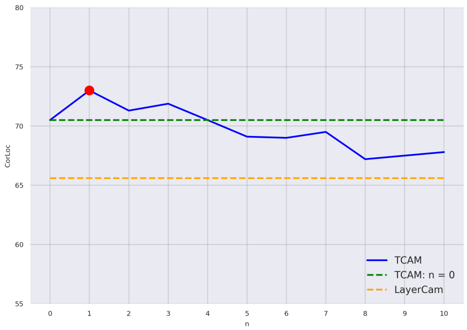

Figure 3: Localization accuracy (`CorLoc`) of TCAM with different temporal dependencies $n$ on the `YTOv1` test set.

**Visual Results.** Fig.4 illustrates prediction cases over labeled ground truth frames. Our method yields CAMs that tend to cover the entire object with a clear distinction between foreground and background. It deals well with multi-instances, and partially visible objects. The second row shows a concrete case where random sampling prevents overfitting over small and strong ROI (bottom-right), and allows other consistent and discriminative objects to emerge (boat at center) from low activations. Fig.5 shows typical failure of our method. They manifest by over-activation over tiny objects, and spill to background. This is mainly caused by heavy presence of wrong ROI activations over the baseline CAM used to generated pseudo-labels. Unfortunately, dominant erroneous pseudo-labels in this case can lead to wrong localization in our method. Their early detection in the baseline CAM and dealing with them is of paramount importance for future extensions of this work. Such issues go under learning with noisy labels which is still an ongoing active domain [65]. This highlights the dependency of our method to the accuracy of the backbone CNN classifier, and baseline CAM.

## 5. Conclusion

CAM-based methods have seen large success in still images for WSOL task. Due to several limitations in current work in WSVOL task, we propose to leverage CAMs for this task. However, since CAMs are not designed to benefit from temporal information in videos, we propose a new module, CAM-TMP, that allows CAMs to do so. It aims to collect available ROI from a sequence of CAMs, which are used to generate pseudo-labels for training. Combined with local and global constraints, we are able to train our model for WSVOL task. Evaluated on two public benchmarks for unconstrained videos, we demonstrated that simple CAM-methods can yield competitive results. Our method yielded new state-of-the-art localization performance. Our ablations show that localization improvement in our method can be done by leveraging short time dependency. Demonstrative videos suggest that our proposal can be easily adapted for subsequent tasks such as video object tracking and detection.

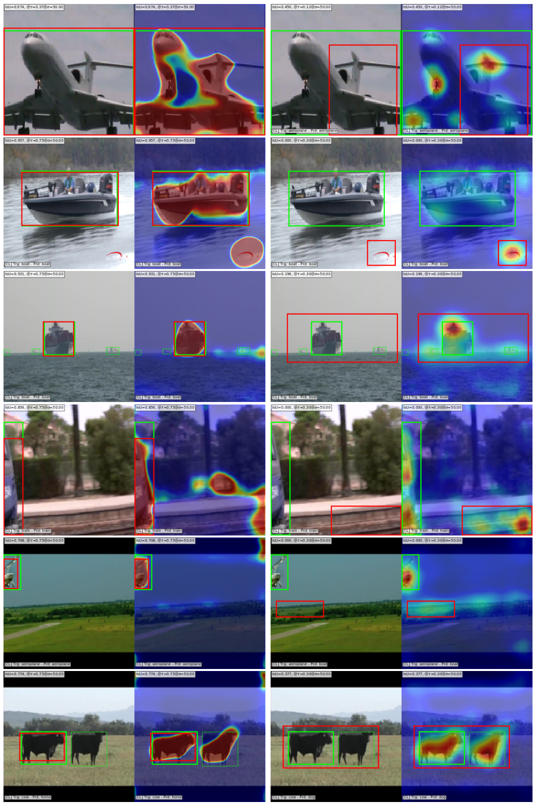

Figure 4: Prediction examples of test sets frames. *Left*: TCAM (ours). *Right*: baseline CAM method, LayerCAM [31]. *Bounding box*: ground truth (green), prediction (red). Second column is predicted CAM over image.

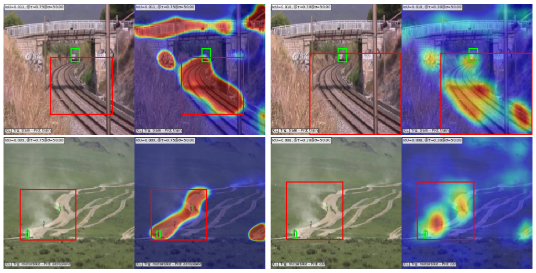

Figure 5: Typical failed cases of our method over test sets. *Left*: TCAM (ours). *Right*: baseline CAM method, LayerCAM [31]. *Bounding box*: ground truth (green), prediction (red). Second column is predicted CAM over image.

## 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).

# References

[1] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *NeurIPS*, 2018.

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

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

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

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

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

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

[8] P. Bergmann, T. Meinhardt, and L. Leal-Taixé. Tracking without bells and whistles. In *ICCV*, 2019.

[9] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

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

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

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

[13] D.-J. Chen, H.-T. Chen, and L.-W. Chang. Video object cosegmentation. In *Multimedia Conference*, 2012.

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

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

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

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

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

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

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

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

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

[23] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. In *BMVC*, 2020.

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

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

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

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

[28] K. He, X. Zhang, S.g Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[29] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

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

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

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

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

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

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

[36] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. *The (Un)reliability of Saliency Methods*, pages 267–280. 2019.

[37] Y. J. Koh, W.-D. Jang, and C.-S. Kim. POD: discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In *CVPR*, 2016.

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

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

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

[41] K. Li, Z. Wu, K.-C. Peng, et al. Tell me where to look: Guided attention inference network. In *CVPR*, 2018.

[42] M. Lin, Q. Chen, and S. Yan. Network in network. *coRR*, abs/1312.4400, 2013.

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

[44] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim. Multiple object tracking: A literature review. *Artificial Intelligence*, 293:103448, 2021.

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

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

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

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

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

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

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

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

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

[54] A. Papazoglou and V. Ferrari. Fast object segmentation in unconstrained video. In *ICCV*, 2013.

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

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

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

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

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

[60] Jérôme Rony, Soufiane Belharbi, Jose Dolz, Ismail Ben Ayed, Luke McCaffrey, and Eric Granger. Deep weakly-supervised learning methods for classification and localization in histology images: a survey. *CoRR*, abs/1909.03354, 2022.

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

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

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

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

[65] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[66] O. Stretcu and M. Leordeanu. Multiple frames matching for object discovery in video. In *BMVC*, 2015.

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

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

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

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

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

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

[73] Saqib Umer, Hassan Dawood, Muhammad Haroon Yousaf, Hussain Dawood, and Haseeb Ahmad. Efficient foreground object segmentation from video by probability weighted moments. *Optik*, 229:166251, 2021.

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

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

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

[77] C. Xu, C. Xiong, and J. J. Corso. Streaming hierarchical video segmentation. In *ECCV*, 2012.

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

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

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

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

[82] C.-L. Zhang, Y.-H. Cao, and J. Wu. Rethinking the route towards weakly supervised object localization. In *CVPR*, 2020.

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

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

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

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

[87] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T.S. Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018.

[88] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T.S. Huang. Self-produced guidance for weakly-supervised object localization. In *ECCV*, 2018.

[89] X. Zhang, Y. Wei, and Y. Yang. Inter-image communication for weakly supervised localization. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, Lecture Notes in Computer Science, 2020.

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

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

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

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