TCAM: Temporal Class Activation Maps for Object Localization in Weakly-Labeled Unconstrained Videos (#1703)

Soufiane Belharbi[1*] & Ismail Ben Ayed[1] & Luke McCaffrey[2] & Eric Granger[1]

[1]LIVIA, Dept. of Systems Engineering, ETS Montreal, Canada
[2]Rosalind and Morris Goodman Cancer Research Centre, Department of Oncology, McGill University

Context

Localization in unconstrained videos is challenging due to: (1) moving objects, (2) camera motion, (3) viewpoint changes, (4) decoding artifacts, (5) editing effects, and (6) costly annotation

Weak supervision: global video tags (classes) are available

State-of-the-art methods in weakly-supervised video object localization have good performance, but:

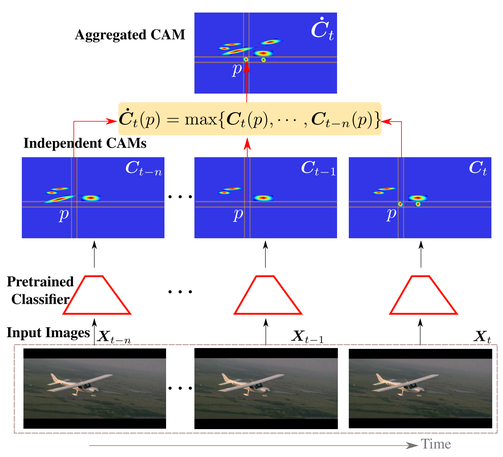
- ☞ multiple sequential and independent stages
- ☞ video tags (labels) are used only to cluster video
- ☞ ROI are not necessarily discriminative
- ☞ motion cues (optical flow) are noisy, not always discriminative, and need post-processing
- ☞ requires solving an optimization problem at inference time: slow inference (build a model per class).

Proposal: Use CAMs with Temporal Dependency

- CAM methods are successful in developing discriminative WSOL models for CNNs
- We adapt CAM methods to exploit the spatiotemporal dependency in videos
- Leverage the slight variations in sets of consecutive frames
- The **CAM-TMP** module aggregates diversified CAMs from n frames

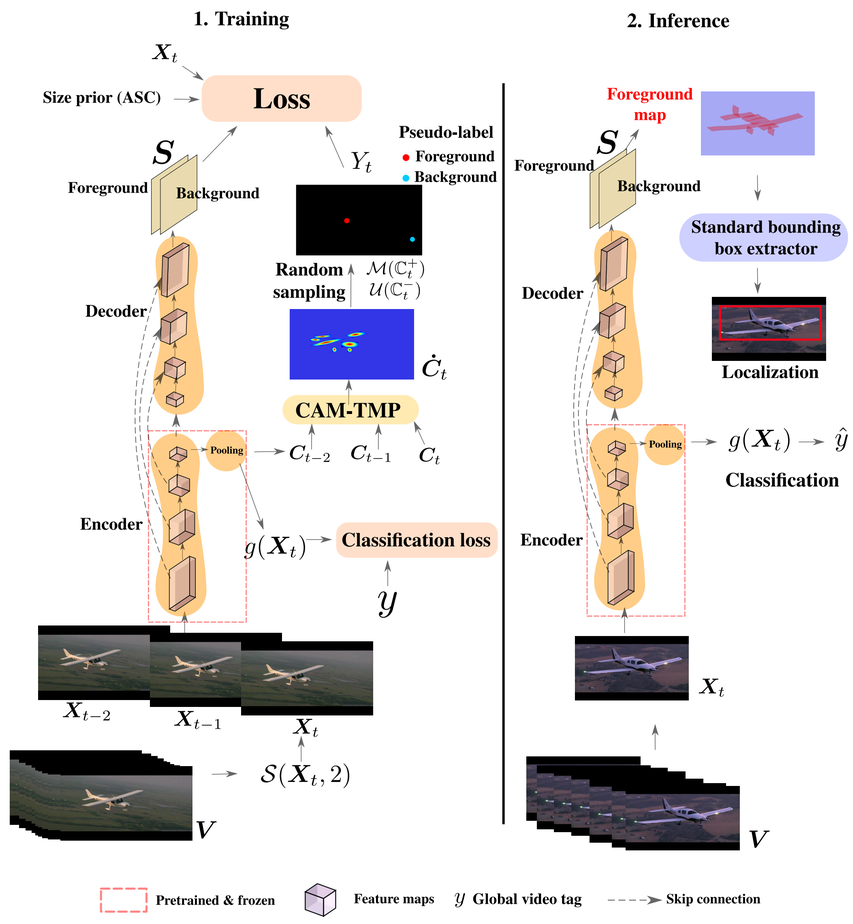
CAM-TMP

CAM Temporal Max-Pooling



Proposal: Overall Architecture

- Includes a learnable decoder to produce accurate CAMs
- Training: accounts for spatio-temporal dependency at a CAM level – it leverages sequences of n frames
- Uses aggregated CAMs to sample pixel pseudo-labels for training the decoder
- Fast inference: CAMs produced from independent frames



Proposal: Training Loss

Total loss: pseudo labels, CRF, unsupervised size constraint.

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

where the classifier is frozen, $\sum_{p \in \Omega_t'} \boldsymbol{H}_p(Y_t, \boldsymbol{S}_t)$ is partial cross-entropy loss over pixel pseudo-labels, $\mathcal{R}(\boldsymbol{S}_t, \boldsymbol{X}_t)$ is a CRF loss, and $\sum \boldsymbol{S}_t^r \geq 0$ is unsupervised size constraint (the Absolute Size Constraints, ASC).

Empirical Results

Datasets: Localization accuracy (`CorLoc`) on YouTube-Object v1.0 and v2.2 datasets

Comparison with State-of-Art Methods

Dataset	Method *(venue)*	Aero	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Avg	Time/Frame
YTOv1	Prest et al. *(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
	Papazoglou et al. *(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
	Joulin et al.*(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
	Kwak et al. *(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
	Rochan et al. *(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
	Tokmakov et al. *(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 *(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
	Tsai et al. *(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
	Haller et al. *(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
	Croitoru et al. (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
	Croitoru et al.(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
	Croitoru et al. (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
	Croitoru et al. (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) *(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) *(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 *(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 *(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 *(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++ *(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++ *(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 *(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 *(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	Haller et al. *(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
	Croitoru et al. (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
	Croitoru et al. (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
	Croitoru et al. (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
	Croitoru et al. (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
	Croitoru et al. (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 *(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 *(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++ *(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++ *(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 *(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 *(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

Visual Results

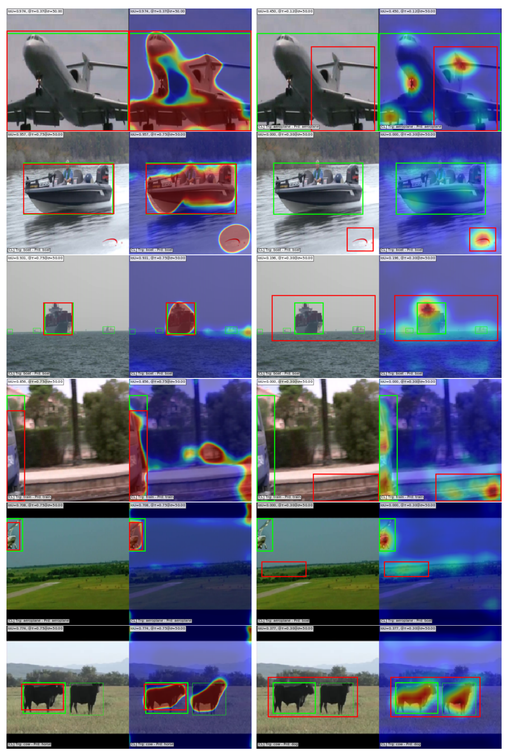


Figure 1. Prediction examples of test sets frames. Left: TCAM (ours). Right: baseline CAM method, LayerCAM Bounding box: ground truth (green), prediction (red). The second column is predicted CAM on images.

Ablation Studies

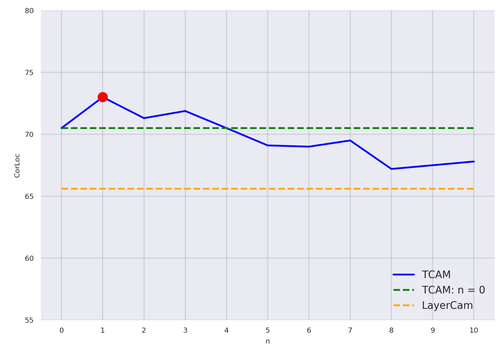


Figure 2. Localization accuracy of TCAM with different temporal dependencies n on the `YTOv1` test set.

Methods		`CorLoc`
Layer-CAM *(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 1. Localization accuracy of TCAM with different losses on the `YTOv1` test set.

Main Conclusions

- Standard CAM methods: can yield discriminative CNNs with accurate localization
- With TCAM: leveraging temporal information during training yielded new state-of-the-art results

Code + video demo: https://github.com/sbelharbi/tcam-wsol-video

soufiane.belharbi.1@ens.etsmtl.ca