

Weakly Supervised Gaussian Networks for Action Detection

Basura Fernando
A*AI, A*STAR Singapore

fernando_basura@scei.a-star.edu.sg

Cheston Tan Yin Chet
I2R, A*STAR Singapore

cheston-tan@i2r.a-star.edu.sg

Hakan Bilen
VICO, University of Edinburgh
hbilen@ed.ac.uk

Abstract

Detecting temporal extents of human actions in videos is a challenging computer vision problem that requires detailed manual supervision including frame-level labels. This expensive annotation process limits deploying action detectors to a limited number of categories. We propose a novel method, called WSGN, that learns to detect actions from weak supervision, using only video-level labels. WSGN learns to exploit both video-specific and dataset-wide statistics to predict relevance of each frame to an action category. This strategy leads to significant gains in action detection for two standard benchmarks THUMOS14 and Charades. Our method obtains excellent results compared to state-of-the-art methods that uses similar features and loss functions on THUMOS14 dataset. Similarly, our weakly supervised method is only 0.3% mAP behind a state-of-the-art supervised method on challenging Charades dataset for action localization.

1. Introduction

Action classification (e.g. [2, 5, 10, 15, 17, 30, 37]) is an extensively studied problem in video understanding with important applications in surveillance, human-machine interaction and human behavior understanding. Recent advances in action classification can be attributed to powerful hierarchical learnable feature representations [15, 30], introduction of large video datasets [15, 16], the use of motion information (e.g. optical flow [30]) and 3D convolutions [35]. While recent methods such as [5, 36] have shown to obtain good action classification performance in various benchmarks, a remaining challenges in video understanding is to localize and classify human actions in long untrimmed videos. Recent methods [6, 39] address localization of actions in long videos in a supervised manner and require action labels for each frame. The super-

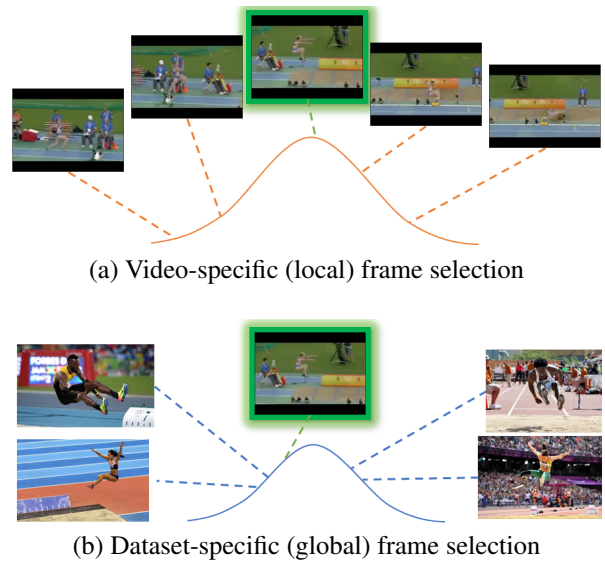


Figure 1. Our method makes use of both video specific (*i.e.* local) and dataset specific (*i.e.* global) score prediction distributions to identify the most relevant set of frames for a given video in a weakly supervised manner for action localization and detection. The likelihood of a frame (shown in green) for an action category (“high-jump”) is obtained by comparing it to the other frames from the same video (local) and frames from other training videos (global) by using two Gaussian normalization functions.

vised paradigm has two shortcomings. First, frame labels are significantly more tedious and expensive to obtain than video-level labels. Second, temporal extent of actions are not as clear as spatial extents of objects (see Figure 2 for an illustration). For instance, Sigurdsson et al. [28] report only 58.7% agreement for temporal boundaries of actions in MultiTHUMOS datasets among human annotators. To address these issues, we propose a novel *weakly supervised* action detection method using only video-level labels. Our method is weakly supervised [40] because it does not use



Figure 2. Example videos for action detection contain “tennis-swing” and “hand-wave” in top and bottom rows respectively. Labeling start and end frames of these actions requires not only a global understanding of these actions but also a local comparison of the candidate frames with their temporal neighbors in the same video.

any frame-level labels during training, however, it outputs accurate frame-level labels at test time.

Our model is trained to **select** and **classify** relevant frames for a given action using video-level labels. It selects relevant frames using a deep neural network named *frame selection module* which acts as a proxy for action localization. At the same time, it classifies relevant frames using another deep neural network, namely *classification module*. Therefore, our model contains two modules (streams), one for frame selection *i.e.*, frame selection module, and one for classifying each frame. Then our model fuses information from both modules to make a video level prediction and trains end-to-end using only video-level labels.

Inspired by the challenging task of predicting temporal boundaries of “hand-wave” action in the bottom row of fig. 2, we hypothesize that accurate action prediction requires not only global understanding of an action class but also a closer look at the frames of a video and local comparison between its frames. We build the frame selection module on this idea such that it assigns a weight to each frame prediction based on both local and global statistics. The local one is realized by a local Gaussian function that picks the most likely frames for an action class and ignores outliers within a video. The predictions that have smaller and larger confidences with respect to the local mean prediction for an action class are normalized (regularized) using a local Gaussian function. This operation is local because the selection involves analysis of predictions among the frames of a single video. This allows us to find the most robust set of predictions for a given action class within a video.

Similarly, the global frame selection strategy involves picking the most relevant frames for an action class from a video by comparing them to global action statistics that are learned over all the videos of that action class. This strategy selects frame predictions that are consistent with globally learned statistics. The analysis of local and global modules are combined to obtain a joint distribution over frames and

action classes (see fig. 1). Finally the outputs of the frame selection and classification modules are combined.

In summary, our contributions are twofold: i) we propose a novel frame selection process for weakly supervised action localization using Gaussian normalization, ii) our Gaussian normalization of scores using both local and global statistics are effective for action localization. Our contributions result in a good improvement in action localization and detection in several challenging benchmarks obtaining results that are competitive with recent weakly supervised techniques.

2. Related Work

Weakly supervised action localization Weakly supervised action classification and localization has been studied in prior work [3, 4, 11, 13, 19, 26, 27, 32]. In [3, 4, 11], the authors use movie scripts to obtain action labels and their approximate temporal boundaries from untrimmed videos and use them as a means of supervision to train action classifiers with various discriminative clustering algorithms. Duchenne *et al.* [11] propose a discriminative learning formulation that simultaneously refines temporal action locations with classifier parameters. Bojanowski *et al.* [3, 4] extend [11] by additionally exploiting the order of actions in a video clip to ensure that the classifier predictions are aligned with the orderings in the scripts.

Extended Connectionist Temporal Classification [13] utilizes weak annotations for action classification by aligning each frame with a label in a recurrent network framework. In contrast to [13] that learns from an ordered list of action labels per video, our method learns to localize action categories from weaker supervision, an unordered set of actions. In principle, such constraints can be incorporated to our learning formulation as constraints. A simple method that implicitly learns to find relevant parts of an object/action after randomly suppressing random parts of images/videos is presented by Singh *et al.* [32]. While

this method is shown to be useful for preventing the network to focus only on discriminative segments, the final model does not achieve a better action classification performance. A more effective weakly supervised action detection method that directly predicts the action boundaries using outer-inner-contrastive loss to parameterize classification loss in terms of temporal boundaries is presented by Shou *et al.* [26]. Nguyen *et al.* [19] propose a loss function comprised of two terms that minimize the video-level action classification error and enforce the sparsity of the segment selection.

Recently Paul *et al.* [20] proposed to employ an attention-based mechanism to select relevant frames and apply pairwise video similarity constraints in a multiple instance framework. Liu *et al.* [18] also utilizes an attention module along with multiple classification streams, each can focus on different discriminative aspects of actions. As a matter of fact, our model also consists of multiple specialized streams, however it differs significantly in terms of temporal modeling functions such as Gaussian and softmax normalization functions to select relevant frames in a weakly supervised manner. We compare to [32, 26, 19, 20, 18] quantitatively in section 4.

Wang *et al.* [38] also employ a two stream method based on [1] for video action detection and localization. Our method differs to Wang *et al.* [38] as our method not only considers local video statistics but also global-dataset-specific score distributions which is crucial for accurate action localization. As also observed in [19], the frame selection mechanism in [38] is limited to select few examples due to the exponential term in softmax operator. While such a mechanism has been shown in [1] to perform well when there are limited instances for object detection, it is not as effective to localize actions which typically comprised of many frames.

Weakly supervised action segmentation [22, 9] is another closely related topic to weakly supervised action detection. It focuses on aligning dense textual descriptions (*e.g.* recipes) such as action transcripts with the frames of the corresponding video (*e.g.* cooking video) by predicting temporal boundaries between different actions. Richard *et al.* [22] use context modeling with context free grammar to perform action segmentation. Ding *et al.* [9] use a temporal convolutional feature pyramid to find coarse ground truth labels and a iterative refinement step using transcripts.

Weakly supervised object localization Our work is also related to the methods in object detection that learn to localize object instances spatially in still images from image-level labels only. The recent work in weakly supervised object detection propose better deep network architectures [1], initialization [33], learning strategies [7] that are less prone to over-fitting, use of various advanced cues such as objectness [8], object size [24] and co-occurrence [25].

3. Problem, approach and model

In this section we present our Weakly Supervised Gaussian Network (WSGN) for action localization. In section 3.1, we present our problem definition, and then in section 3.2 we present a simple naive approach to weakly supervised action localization. Finally, in section 3.3, we present our approach and the methodology.

3.1. Weakly supervised action localization problem

Let $V = \langle I_1, I_2, \dots, I_t, \dots, I_T \rangle$ be a sequence of frames where $I_t \in \mathcal{I} = \mathbb{R}^{3 \times H \times W}$ is the t^{th} frame of the video V . T denotes the video length which vary from video to video. Assume that we are given a set of N training videos and its video-level labels $\{V^i, \mathbf{y}^i\}$ where $\mathbf{y} \in \mathcal{Y} = \{0, 1\}^C$ indicates the presence/absence of human action classes as a C -dimensional ground-truth binary vector for *each training video* V . The q^{th} element of the vector \mathbf{y} is set to one when q^{th} human action is present in the video, otherwise it is set to zero. We wish to learn a function that predicts the presence/absence of human action classes not at video-level but at frame-level for a testing video *i.e.* to predict C -dimensional binary vector \mathbf{y}_t for each frame I_t . The learning becomes weakly supervised as what is being predicted at test time is more detailed than what is used for training [40]. We predict frame label vector (*i.e.* \mathbf{y}_t) at test time for each frame using a model that is trained with *video-level labels* \mathbf{y} . Therefore, our action localization task (*i.e.* predicting \mathbf{y}_t for each frame) is weakly supervised.

Let us denote a trainable feature extractor that returns a M -dimensional vector for each frame by $f(I_t, \theta) : \mathcal{I} \rightarrow \Omega = \mathbb{R}^M$. Here θ are the learned parameters of f . A classification network $h(\cdot, \theta_{cls}) : \Omega \rightarrow \mathcal{Y}$ takes the feature vector $f(I_t, \theta)$ and returns a C -dimensional *action classification score vector*. Here θ_{cls} are the trainable parameters of h . Action classification score vector for frame I_t is then obtained by the joint model $h(f(I_t))$. Next, we present a simple weakly supervised action localization method which we use as a baseline in our experiments.

3.2. Naive weakly supervised action localization

When frame-level action class annotation vectors \mathbf{y}_t are known for the frames of training videos, one can train f and h to minimize binary cross-entropy loss at frame-level. As we assume that no ground truth frame-level labels are available for training, we are limited to use video-level label vectors \mathbf{y} to train our action localization model that predicts \mathbf{y}_t at test time.

A simple strategy to obtain a video-level prediction $\hat{\mathbf{y}}$ from a sequence of frames is to average frame-level predictions over the whole sequence as follows:

$$\hat{\mathbf{y}} = \sum_{t=1}^T \frac{1}{T} \sigma(h(f(I_t, \theta), \theta_{cls})) \quad (1)$$

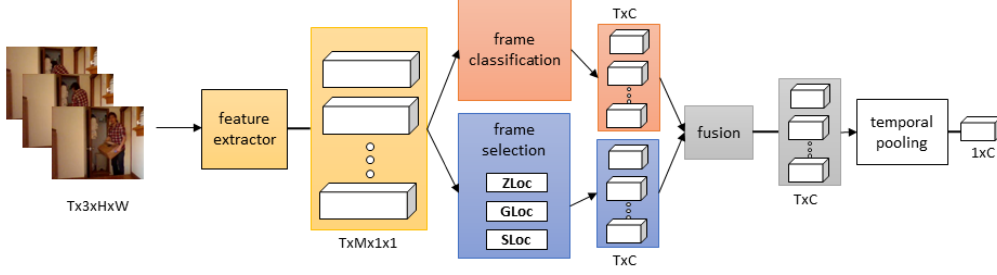


Figure 3. Illustration of our weakly supervised Gaussian action detection framework (WSGN).

where T is the number of frames in the sequence and can vary from video to video, σ is the softmax normalization function over predicted score vector for each frame. To train such a model, we minimize the binary cross entropy loss $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ over predicted probability vector ($\hat{\mathbf{y}}$) and the ground truth. During testing, we use function $h(f(I_t))$ to label each frame. However, this method naively considers an equal importance for each frame to obtain a video level score prediction by simply averaging their scores. We denote this approach the **Naive Weakly Supervised** action localization. Here we hypothesize that a good model should carefully choose the “relevant” frames for the action and then average the scores of only those.

3.3. Our approach

Our approach is to learn another network in addition to the “classification module” (i.e. $\sigma(h(f(I_t, \theta), \theta_{cls}))$) that can identify the relevance of each frame for the task of video action classification. We call this network “frame selection module” and denote it by function $g(I_t, V, \theta_g)$ where θ_g is the learnable parameter vector—see fig. 3. This network function acts as a proxy for action localization and weighs each frame per action class depending on the relevance of the frame to recognize the action.

Similar to h , g function also returns a C -dimensional weight vector for each frame I_t . However, g function differs to the classification module function $\sigma \circ h \circ f$ in two aspects. First, its objective is to find frames that are relevant to each action class. Second, while the classification module h scores each frame independent of other frames, the frame selection module scores each frame relatively by considering frames both from the video V and the entire dataset. We describe the details of the relative scoring functions in the following paragraphs.

The final video-level prediction is obtained by a weighted average of all frame classification predictions where the weights are defined by g as shown in equation 2. Here \odot is the element-wise product between weights and classification probability vectors.

$$\hat{\mathbf{y}} = \frac{1}{T} \sum_{t=1}^T g(I_t, V, \theta_g) \odot \sigma(h(f(I_t, \theta), \theta_{cls})) \quad (2)$$

The video-level prediction $\hat{\mathbf{y}}$ can now be used with a binary cross entropy loss $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ and enables our action localization method to be trained with video-level labels.

During inference, we simply skip this temporal averaging step and instead use $g(I_t, V, \theta_g) \odot \sigma(h(f(I_t, \theta), \theta_{cls}))$ to obtain frame-level predictions and perform action localization by using these scores. In next part, we discuss how to formulate a good g function for the task of action localization.

WSGN: Gaussian frame selection module. Here we explain the frame selection module g which is complementary to the classification module $\sigma(h(f(I_t, \theta), \theta_{cls}))$. To this end, we design the frame selection module in a way that it can predict the relevance of a frame by a comparative analysis to the rest of the frames. In particular g function consists of three components which are responsible for (i) extracting features from sequence of frames, (ii) predicting a C -dimensional score vector for each frame based on the extracted features, (iii) normalizing those score vectors to select frames. For the first component, g shares the feature extractor f with the classification module for computational efficiency. For the second part, g has a dedicated classifier also denoted by $h(\cdot, \theta_{det})$ which takes the feature $f(I_t)$ and returns a C -dimensional *action selection score vector* but parameterized a different set of parameters θ_{det} . As the input and output dimensions of $h(\cdot, \theta_{det})$ are same with $h(\cdot, \theta_{cls})$, we simply use the same function structure $h(\cdot, \cdot)$ for brevity. The frame selection score vector obtained by $h(f(I_t, \theta), \theta_{det})$ is then denoted by \mathbf{x}_t and the class-specific score for the q^{th} action class is then denoted by scalar \mathbf{x}_t^q .

The objective of frame selection module $g(\cdot)$ is to select relevant frames for a given action. To do so we make use of both video-specific and dataset-wide statistical information to find frame-action predictions that are most probable using *normalization functions*. These *normalization functions* compare predicted scores for each frame against other frames to obtain a *likelihood estimate* for each prediction $h(f(I_t), \theta_{det})$. However, the aim here is not to obtain an estimate for the presence of action class q in frame I_t as done by the classification module $h(f(I_t), \theta_{cls})$ but to es-

timate a likelihood of each prediction $h(f(I_t), \theta_{det})$ with respect to local and global score distributions. Higher the likelihood estimate of a prediction with respect to others, a higher weight is given by frame selection module, to the corresponding classification prediction in eq. (2). Using the feature extractor $f(\cdot, \theta)$, frame selection function $h(\cdot, \theta_{det})$ and normalization functions, our frame selection module $g(\cdot)$ finds relevant frames for each action class. Next we describe three *normalization functions* including local (ZLoc), global (GLoc) and softmax (SLoc) that are used with our WSGN method.

ZLoc: Local variant of WSGN The local variant of WSGN model uses a Gaussian normalization function to find weights for each prediction using a likelihood estimate over the frames of a single video. We estimate the likelihood of frame selection scores x_t^q for q^{th} class relative to all other frames within the same video $t = \{1, 2, \dots, T\}$ using a Gaussian likelihood function as follows:

$$z_t^q = \exp(-[\frac{x_t^q - \mu_z^q}{s_z^q}]^2). \quad (3)$$

Here, μ_z^q and s_z^q are the statistical mean and the standard-deviation of all frame selection scores for q^{th} class obtained by $h(f(\cdot, \theta), \theta_{det})$ for video V . In fact z_t^q is a probability estimate of x_t^q with respect to all other frame selection scores within the video. It assigns lower probabilities to those predictions that are very different from the mean prediction. If x_t^q is very large compared to the mean, it is considered as an outlying prediction. This encourages our model to identify not only the most salient frame for the action but also the whole temporal extent of the action.

By using eq. (3), g function assigns a class-specific weight z_t^q to each frame I_t which is further multiplied with the classification prediction (i.e. from $\sigma(h(f(I_t, \theta), \theta_{cls}))$), as indicated in eq. (2). Let us denote this normalization operation by $g_{zloc}(\cdot)$ which takes all action selection score vectors (\mathbf{x}_t for all $t = \{1, 2, \dots, T\}$) and returns a weight vector \mathbf{z}_t for each frame. Then the function $g(I_t, V, \theta_g)$ would return a weight vector \mathbf{z}_t where

$$g(I_t, V, \theta_g) = g_{zloc}(\cdot) \circ h(\cdot, \theta_{det}) \circ f(\cdot, \theta)(I_t, V). \quad (4)$$

The $g_{zloc}(\cdot)$ function does not have any learnable parameters, thus learning the local variant of our method (denoted by ZLoc or Z_{loc}) involves optimizing three sets of parameters θ , θ_{cls} and θ_{det} using the binary cross-entropy loss.

GLoc: Global variant of WSGN. While ZLoc variant of our WSGN method considers statistics from frames of a single video to normalise scores, the global variant one **GLoc** compares each frame frame selection score x_t^q with the frames from all the training videos. As a direct comparison to all frames is computationally expensive and not even possible within the memory of the standard GPUs, we

instead choose to use dataset-wide or global statistics with a Gaussian function per action category over the frame selection scores. To this end we propose to learn a mean vector (μ_l^q) and standard deviation vector (s_l^q) per action category jointly along with the other network parameters. The subscript l of μ_l^q is used to indicate that they are learned but not statistically computed over the scores. Both μ_l^q and s_l^q are learned using the training samples and therefore, representative of the global dataset specific information. Our new GLoc normalization operation is then given by eq. (5).

$$l_t^q = \exp(-[\frac{x_t^q - \mu_l^q}{s_l^q}]^2) \quad (5)$$

The associated normalization function is denoted by $g_{gloc}(\cdot)$ which takes the class selection score vector \mathbf{x}_t as input. The weight vector returned by global GLoc approach is denoted by $\mathbf{l}_t = g(I_t, \theta_g)$ where

$$g(I_t, \theta_g) = g_{gloc}(\cdot, \mu_l, s_l) \circ h(\cdot, \theta_{det}) \circ f(\cdot, \theta)(I_t). \quad (6)$$

In contrast to local Gaussian approach (ZLoc), in GLoc, we learn parameter vectors μ_l and s_l in addition to θ , θ_{cls} and θ_{det} . The weight \mathbf{l}_t is estimated not from a single video but all the training samples. If the frame selection score x_t^q is more likely w.r.t. global score distribution, then l_t^q will be higher and the prediction from the classification path $\sigma(h(f(I_t, \theta), \theta_{cls}))$ for class q is highly weighted.

SLoc: Softmax variant of WSGN. For completeness, we also propose to use a commonly used normalization function, softmax but apply it to normalize the video specific scores x_t^q over the frames of a video but not over the feature channels so that the sum of the frame selection scores are normalized to 1 for a video:

$$s_t^q = \frac{e^{x_t^q}}{\sum_{i=1} e^{x_i^q}} \quad (7)$$

The associated normalization function is then denoted by $g_{sloc}(\cdot)$.

WSGN: Complete model. We make use of all three normalization function, namely the local ZLoc, global GLoc and softmax-based SLoc variants in our WSGN. To integrate the predictions from three streams, we propose a simple averaging strategy, i.e. $g = \text{avg}(g_{zloc}(\cdot), g_{gloc}(\cdot), g_{sloc}(\cdot)) \circ h(\cdot, \theta_{det}) \circ f(\cdot, \theta)$, where avg denotes element-wise averaging over three normalization functions. The combined class-specific frame selection weight for frame I_t can simply be obtained by the average of weights i.e. $\frac{1}{3}(z_t^q + l_t^q + s_t^q)$. Now we can finally combine the predictions of classification module denoted by $\sigma(h(f(I_t, \theta), \theta_{cls}))$ and the frame selection module by $g(\cdot)$. A visual illustration of our method is shown in fig. 3. For action detection and localization, we use the score returned by $g(I_t, V, \theta_g) \odot \sigma(h(f(I_t, \theta), \theta_{cls}))$ for each frame.

4. Experiments

4.1. Datasets

We evaluate our **WSGN** method on two standard action localization benchmarks, namely the Charades [29] and THUMOS14 [14].

Charades [29] is composed of 9,848 indoor videos with an average length of 30 seconds, comprising 157 action classes from 267 different people that are recorded in their homes and performing everyday activities. Each video is annotated with action labels and duration which allow evaluation for action localization. We use a standard evaluation procedure introduced in [27] for action localization using fixed train (7985) and validation (1863) splits. As done in [27], we predict a score for 157 classes for 25 equally spaced time-points in each video and then report action localization mAP.

THUMOS14 [14] dataset consists of very long videos (average length is 3.5 minutes) having 20 human action classes for action detection task. In this dataset, we follow the evaluation procedure in the previous work [32, 26] to provide a fair comparison. Concretely, we use the validation set containing 200 untrimmed videos for training and evaluate our model on the test set containing 213 videos.

4.2. Implementation details

Features and networks: We use two convolutional neural networks; namely the VGG16 [31] and ResNet34 [12] that are pretrained for ImageNet classification task [23] for Charades dataset. These are trained end-to-end. We use ImageNet pre-trained I3D (**I3D-I**) and UntrimmedNet [38] features for THUMOS14 experiments. Furthermore, we evaluate our methods using I3D networks [5] that is pretrained for video action classification on Kinetics dataset [16] (denoted by (**I3D-K**)) for both THUMOS14 and Charades experiments to obtain competitive state-of-the-art results. These are only fine-tuned and not trained end-to-end. For all these networks, we take the output before the classification layer and include a dropout layer with a dropout rate of 0.5 for image classification networks and a dropout rate of 0.8 for I3D networks. These serve as our feature extractor network $f(\cdot, \theta)$. As the feature classification networks (*i.e.* $h(\cdot, \theta_{cls})$ and $h(\cdot, \theta_{det})$), we use a simple two layered neural network to produce classification scores (hidden layer size is set to the input feature size).

Training details: We use a learning rate of 10^{-4} for ResNet34, and I3D and a learning rate of 10^{-3} for VGG and trained for a maximum of 80 epochs. We use a batch size of 128 videos and 32 sub-batches and a weight decay of 0.0005. Because some videos are very long, we sample every 5th frame and perform a temporal data augmentation (vary the start of the sampled sequence from 1st to 15th frame). Only for THUMOS14, during inference, we use all

	VGG16	Resnet34	I3D-K
Supervised	9.0	10.1	18.7
Naive W. Sup. (eq. (1))	5.2	5.2	13.7
WSGN - SLoc	6.0	7.1	14.9
WSGN - ZLoc	8.7	9.0	16.8
WSGN - GLoc	8.5	9.3	17.2
WSGN - SLoc + GLoc	8.7	9.4	18.2
WSGN - ZLoc + GLoc	8.8	9.4	18.2
WSGN - Complete	8.9	9.7	18.3

Table 1. Action localization performances in mAP(%) for supervised and weakly supervised methods with different base networks on Charades by using only RGB input. WSGN - SLoc: Softmax normalization, WSGN - ZLoc: local Gaussian normalization, WSGN - GLoc: global parametric Gaussian normalization.²

frames, however we set the mini-batch size to one to make sure we fit videos in GPU memory of (4×16GB). We use standard, data augmentation at video frame level (flipping, random cropping, etc.) but apply the same augmenting operation for the entire video to obtain a temporally smooth video after data augmentation.

4.3. Ablation study on Charades dataset.

In this section we compare several baselines and variants of weakly supervised models presented in section 3 on Charades dataset. We experiment with three network architectures, namely VGG16, Resnet34 and I3D-K. For I3D-K, we use bi-linear interpolation to obtain frame-wise feature representation and fine-tune with video level annotations.

We analyze the impact of different normalization functions *i.e.* ZLoc, GLoc and SLoc of our WSGN. We compare our method with (1) RGB-based naive weakly supervised baseline (**Naive**) which corresponds to eq. (1). (2) We also report results for weakly supervised action localization only with Softmax normalization denoted by **SLoc**, (3) weakly supervised action localization only with local Gaussian normalization denoted by **ZLoc** (4) weakly supervised action localization only with global parametric Gaussian normalization denoted by **GLoc**. Our complete model shown in Figure 3 is denoted by **WSGN-Complete**. We also report results for fully **Supervised** case where we train a model using frame level annotations. In this case, we minimize the a combinations of losses $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) + \sum_t \frac{1}{T} \mathcal{L}(\mathbf{y}_t, \hat{\mathbf{y}}_t)$ where $\hat{\mathbf{y}} = \frac{1}{T} \sum_t \sigma(h(f(I_t, \theta), \theta_{cls}))$ and $\hat{\mathbf{y}}_t = \sigma(h(f(I_t, \theta), \theta_{cls}))$. Results are shown in Table 1.

Several interesting observations can be listed based on these results. First, we obtain considerable improvements in action localization using our WSGN-Complete method over naive approach (improvement of **3.7** mAP for VGG16, **4.5** mAP for Resnet34, and **4.6** mAP for I3D). Interestingly, our method seems to gain more when much richer network architectures are used as the best improvement over Naive Weak Supervision (eq. (1)) is obtained with I3D network. Secondly, as an individual method, global Gaussian-

based WSGN-GLoc seems the most effective one. Local Gaussian-based normalization method (ZLoc) is also as effective as GLoc method. Because SLoc uses softmax selection, its output is sparse and thus it could select only few important frames rather than entire coverage of an action. This leads to relatively low performance for SLoc. Combination of both ZLoc and GLoc is considerably more effective. Combination of all normalization methods seems to be the most beneficial in this dataset for action localization. Our weakly supervised results are surprisingly as good as the fully supervised method on all three network architectures indicating the effectiveness of our novel frame selection module. We conclude that WSGN weakly supervised action localization is effective on Charades dataset across wide variety of base network architectures.

4.4. Ablation study on THUMOS14.

In this section we evaluate our method on THUMOS14 dataset using ImageNet pretrained I3D (I3D-I). We extract features from both RGB and optical flow streams following the protocol used in [5] and then use bi-linear interpolation to obtain frame-wise feature representation and fine-tune with video level annotations. We evaluate varying the detection IoU threshold using the standard evaluation protocol as in [14]. As our method is only weakly supervised, to generate action detection boundaries *i.e.* start and end of each action, we make use of two heuristics. First, we threshold scores to find candidate frames and then generate candidate segments using those consecutive frames that has a score greater than the threshold. It should be noted, because we use local/global Gaussian normalization, returned scores are already managed to get rid of outliers and very small scores. Secondly, we use only those candidate segments that are longer than one second for evaluation. We report results in Table 2.

We observe a similar trend to Charades dataset where all variants of our WSGN is effective than Naive Weak Supervision (eq. (1)) method. Interestingly, even if our supervised results are far better than our weakly supervised method, obtained results are very encouraging. The improvement we obtain over Naive Weak Supervision (eq. (1)) method is comparatively greater than the gap between supervised performance and our WSGN (SLoc + ZLoc + GLoc). We conclude that our method is very effective for weakly supervised action detection in THUMOS14 dataset.

4.5. Comparison to prior state-of-the art.

We compare with several weakly supervised action detection methods that have been evaluated on THUMOS14 dataset [34, 32, 38, 26, 19, 20, 18] in Table 3. In particular, some very successful recent methods such as STPN [19], W-TALC [20] and Comp [18] use I3D pretrained on Kinetics (I3D-K). Therefore, we report results with I3D-K

Method	0.1	0.2	0.3	0.4	0.5
Supervised	59.7	51.9	47.4	40.1	32.8
Naive W. Sup. (eq. (1))	42.9	36.6	28.0	20.9	14.2
WSGN - SLoc	45.3	38.9	31.0	23.5	16.3
WSGN - ZLoc	54.5	47.8	38.8	28.9	20.0
WSGN - GLoc	45.2	38.8	30.2	22.4	14.6
WSGN - SLoc + GLoc	48.4	42.5	34.1	26.0	18.0
WSGN - ZLoc + GLoc	54.7	48.6	39.4	29.4	20.7
WSGN - Complete	55.3	47.6	38.9	30.0	21.1
Gap	4.4	4.3	8.5	10.1	11.7
Improvement	12.4	11	10.9	9.1	6.9

Table 2. Action detection performance on THUMOS14 dataset using I3D-I for variants of our weakly supervised WSGN action detection method. We change IoU threshold from 0.1 to 0.5 and report results. WSGN - SLoc: Softmax normalization, WSGN - ZLoc, local Gaussian normalization, WSGN - GLoc: global parametric Gaussian normalization. The Gap between the supervised results and our WSGN (SLoc + ZLoc + GLoc) is shown. The improvement obtained over Naive Weak Supervision (eq. (1)) method is also shown.

Method	0.1	0.2	0.3	0.4	0.5	0.6	0.7
Supervised faster-rcnn [6]	59.8	57.1	53.2	48.5	42.8	33.8	20.8
LAF [34]	12.4	11.0	8.5	5.2	4.4	—	—
Hide-and-seek [32]	36.4	27.8	19.5	12.7	6.8	—	—
UntimmedNets [38]	44.4	37.7	28.2	21.1	13.7	—	—
AutoLoc [26]	—	—	35.8	29.0	21.2	13.4	5.8
STPN [19] (I3D-K)	52.0	44.7	35.5	25.8	16.9	9.9	4.3
W-TALC [20] (I3D-K)	55.2	49.6	40.1	31.1	22.8	—	7.6
Comp [18] (I3D-K)	57.4	50.8	41.2	32.1	23.1	15	7.0
Our (I3D-K)	55.3	48.8	37.2	30.2	21.1	13.8	8.2
Our+CASL (I3D-K)	57.9	51.2	42.0	33.1	25.1	16.7	8.9
W-TALC [20] (UNTF)	49.0	42.8	32.0	26.0	18.8	—	6.2
WSGN (UNTF)	51.1	44.4	34.9	26.3	18.1	11.6	6.5

Table 3. Action detection performance on THUMOS14 dataset for various weakly supervised state-of-the-art methods. We also show results for supervised state-of-the-art method [6] as a reference. We also compare with s.o.a. W-TALC [20] with UntimmedNet [38] features (UNTF).

and UntimmedNet [38] features (UNTF). Additionally, we make use of CASL loss presented in [20] to further improve our results. Our method trained with the CASL loss outperforms all other methods. Also our method without CASL loss is better than recent effective methods such as STPN [19] which relies on feature attention over frame features similar to us. Indeed, the idea of co-activity loss presented in W-TALC [20] is effective and complimentary to our method. In-fact, state-of-the art methods such as recently presented [18] might be complimentary to us as well. We leave the use of diversity loss [18] for future work. We obtain better results with UNTF features compared to [20]. Especially, for IoU of 0.1, our method is only **1.9 mAP** behind state-of-the-art faster-rcnn-temporal-localization [6]. However, supervised methods perform way better than weakly supervised methods for larger IoU thresholds. This is not surprising as the task becomes difficult with larger IoUs. Nevertheless, we obtain somewhat

	Method	Network	Input	Loc.
Supervised	Temporal Fields [27]	VGG16	RGB	9.0
	Two Stream++ [30]	VGG16	RGB+OF	10.9
	Temporal Fields [27]	VGG16	RGB+OF	12.8
	Super-Events [21]	I3D	RGB	18.6
	Super-Events [21]	I3D	RGB+OF	19.4
Wk. Sup.	WSGN (ours)	VGG16	RGB	8.9
	WSGN (ours)	ResNet34	RGB	9.7
	WSGN (ours)	I3D	RGB	18.3

Table 4. Comparison to the state-of-the action localization methods on Charades dataset. All other methods we compare with are fully supervised in which frame level annotations of Charades dataset are at disposal during the training process. Our method is only “Weakly Supervised” (WS) and obtain competitive results.

encouraging results even compared to state-of-the-art supervised methods for smaller IoU thresholds while obtaining some encouraging results even for very large IoU of 0.7–see Table 3.

To the best of our knowledge, no prior weakly supervised method have evaluated on challenging Charades dataset for action localization. Therefore, we only compare with supervised methods, which actually used frame level annotations. Results are reported in Table 4. Notably, our method obtains competitive results compared to fully supervised methods using both VGG16 and I3D architectures. Effective Temporal Fields [27] method obtains 9.0 mAP using VGG16 and RGB stream while our weakly supervised method is slightly worse (8.9 mAP). Similarly, our WSGN with I3D feature extractor performs only **0.3 mAP** worse than supervised Super-Events [21] method. This is an indication of the impact of our novel Gaussian normalization-based frame selection module. However, our results are 1.1 mAP lower than Super-Events [21] when used with both RGB and optical flow (although we don’t use optical flow for Charades dataset). We conclude that our weakly supervised method is effective for video action localization (Charades dataset) and detection (THUMOS14 dataset).

4.6. Analysis of localization components.

Our WSGN method has several computational outputs that predicts categorical information. For example the action classification score $h(\cdot, \theta_{cls}) \circ f(\cdot, \theta)$, and action selection score $h(\cdot, \theta_{det}) \circ f(\cdot, \theta)$. The global normalization function outputs $g_{gloc}(\cdot)$, the local normalization outputs $g_{zloc}(\cdot)$ and softmax normalization outputs $g_{sloc}(\cdot)$ also returns C-dimensional weights. Overall, the frame selection module $g(\cdot)$ outputs C-dimensional weight vector for each frame by taking average of all normalization functions. Finally, both classification and frame selection module $g(\cdot)$ outputs are multiplied to get the frame-wise final prediction which we use for action localization. We perform an analysis on these outputs and report action detection performance in Figure 4 using THUMOS14 dataset.

First, we see that all normalization outputs do not per-

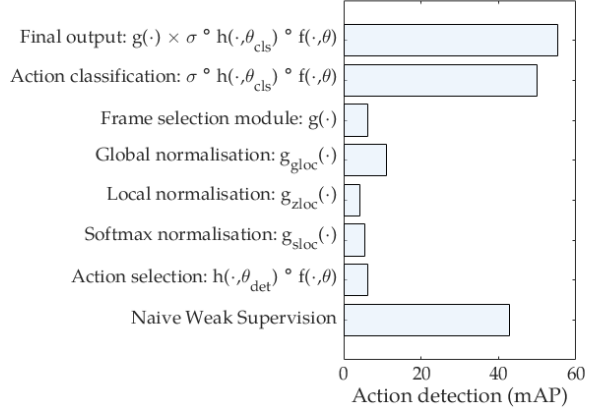


Figure 4. Dissecting the model outputs for action detection.

form as good as classification output $h(\cdot, \theta_{cls}) \circ f(\cdot, \theta)$. This is not surprising as the goal of normalization is simply frame selection for each action. However, the classification module outputs performs satisfactorily and already obtains better results than Naive Weak Supervision (eq. (1)) method indicating in-fact the addition of frame selection module itself helped to improve the classification parameters as well. This indirectly indicates that, our frame selection module $g(\cdot)$ help to improve the video representation $f(\cdot, \theta)$. The best results are obtained by the final output indicating the advantage of having two separate functions for frame selection and classification.

5. Conclusion

In this paper we propose a weakly supervised action localization method where both frame selection and classification are learned jointly with a deep neural network in an end-to-end manner. We show that accurate action localization require both video-level and dataset-wide frame comparison. As demonstrated in our experiments, combination of both local and global strategies result in better performance and obtains state-of-the-art results in two challenging datasets. Importantly our method further narrows down the gap between the supervised and weakly supervised paradigms. For future work we plan to extend our weakly supervised localization method from temporal to spatio-temporal domain by exploring higher dimensional normalization strategies.

Acknowledgement: This research was supported by the National Research Foundation, Singapore (Award Number: NRF2015-NRF-ISF001-2451) and the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

- [1] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016. 3
- [2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *TPAMI*, 23(3):257–267, 2001. 1
- [3] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, pages 628–643. Springer, 2014. 2
- [4] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, and C. Schmid. Weakly-supervised alignment of video with text. In *ICCV*, pages 4462–4470, 2015. 2
- [5] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *CVPR*, 2017. 1, 6, 7
- [6] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, pages 1130–1139, 2018. 1, 7
- [7] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *TPAMI*, 39(1):189–203, 2017. 3
- [8] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, pages 452–466. Springer, 2010. 3
- [9] L. Ding and C. Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, 2018. 3
- [10] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005. 1
- [11] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, pages 1491–1498, 2009. 2
- [12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *ICCV*, pages 770–778, 2016. 6
- [13] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, pages 137–153. Springer, 2016. 2
- [14] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. <http://crcv.ucf.edu/THUMOS14/>, 2014. 6, 7
- [15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1
- [16] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 6
- [17] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003. 1
- [18] D. Liu, T. Jiang, and Y. Wang. Completeness modeling and context separation for weakly supervised temporal action localization. In *CVPR*, June 2019. 3, 7
- [19] P. Nguyen, T. Liu, G. Prasad, and B. Han. Weakly supervised action localization by sparse temporal pooling network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 3, 7
- [20] S. Paul, S. Roy, and A. K. Roy-Chowdhury. W-talc: Weakly-supervised temporal activity localization and classification. In *ECCV*, pages 563–579, 2018. 3, 7
- [21] A. Piergiovanni and M. S. Ryoo. Learning latent super-events to detect multiple activities in videos. In *CVPR*, 2018. 8
- [22] A. Richard, H. Kuehne, and J. Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *CVPR*, 2018. 3
- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 6
- [24] M. Shi and V. Ferrari. Weakly supervised object localization using size estimates. In *ECCV*, pages 105–121. Springer, 2016. 3
- [25] Z. Shi, P. Siva, and T. Xiang. Transfer learning by ranking for weakly supervised object annotation. *arXiv preprint arXiv:1705.00873*, 2017. 3
- [26] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *ECCV*, pages 162–179, 2018. 2, 3, 6, 7
- [27] G. A. Sigurdsson, S. Divvala, A. Farhadi, and A. Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017. 2, 6, 8
- [28] G. A. Sigurdsson, O. Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017. 1
- [29] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, pages 510–526. Springer, 2016. 6
- [30] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014. 1, 8
- [31] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6
- [32] 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. 2, 3, 6, 7
- [33] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. In *ICML*, 2014. 3
- [34] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia. Temporal localization of fine-grained actions in videos by domain transfer from web images. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 371–380. ACM, 2015. 7
- [35] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *ICCV*, 2015. 1

- [36] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. [1](#)
- [37] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. [1](#)
- [38] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proc. CVPR*, 2017. [3](#), [6](#), [7](#)
- [39] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2914–2923, 2017. [1](#)
- [40] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017. [1](#), [3](#)