Action Recognition from Single Timestamp Supervision in Untrimmed Videos

Davide Moltisanti
Visual Information Lab
University of Bristol
davide.moltisanti@bristol.ac.uk

Sanja Fidler
University of Toronto
NVIDIA Vector Institute
fidler@cs.toronto.edu

Dima Damen
Visual Information Lab
University of Bristol
dima.damen@bristol.ac.uk

Abstract

Recognising actions in videos relies on labelled supervision during training, typically the start and end times of each action instance. This supervision is not only subjective, but also expensive to acquire. Weak video-level supervision has been successfully exploited for recognition in untrimmed videos, however it is challenged when the number of different actions in training videos increases. We propose a method that is supervised by single timestamps located around each action instance, in untrimmed videos. We replace expensive action bounds with sampling distributions initialised from these timestamps. We then use the classifier's response to iteratively update the sampling distributions. We demonstrate that these distributions converge to the location and extent of discriminative action segments.

We evaluate our method on three datasets for fine-grained recognition, with increasing number of different actions per video, and show that single timestamps offer a reasonable compromise between recognition performance and labelling effort, performing comparably to full temporal supervision. Our update method improves top-1 test accuracy by up to 5.4%. across the evaluated datasets.

1. Introduction

Typical approaches for action recognition in videos rely on full temporal supervision, i.e. on the availability of the action start and end times for training. When the action boundaries are available, all (or most of) the frames enclosed by the temporal bounds can be considered relevant to the action, and thus state-of-the-art methods randomly or uniformly select frames to represent the action and train a classifier [30, 12, 33, 6, 34, 16]. Collecting these boundaries is not only notoriously burdensome and expensive, but also potentially ambiguous and often arbitrary [21, 29, 7].

With an increasing need for bigger video datasets, it is important to scale up the annotation process to foster more rapid advance in video understanding. In this work, we attempt to alleviate such annotation burden, using *single* roughly aligned timestamp annotations in untrimmed videos - i.e. videos labelled with only one timestamp per action, located close to the action of interest. Such labelling is quicker to collect, and importantly is easier to communicate to annotators who do not have to decide when the action starts or ends, but only label one timestamp within or close to the action. Single timestamps can alternatively be collected from audio narrations and video subtitles [8, 1].

To utilise this weak supervision, we propose a sampling distribution, initialised from the single timestamps, to select relevant frames to train an action recognition classifier. Due to the potential coarse location of the timestamps, and to actions having different lengths, the initial sampling distributions may not be well aligned with the actions, as showed in Figure 1 (top). We thus propose a method to update the parameters of the sampling distributions during training, using the classifier's response, in order to sample more relevant frames and reinforce the classifier (Figure 1, bottom).

Our attempt is inspired by similar approaches for single point annotations in image based semantic segmentation [2], where results achieved using such point supervision have slightly lower accuracy than those obtained with fully annotated masks, but outperform results obtained with image-level annotations. Correspondingly, we show that single timestamp supervision for action recognition outperforms video-level supervision.

We test our method on three datasets [15, 9, 8], of which [8] is annotated with single timestamps from live audio commentary. We show that our update method converges to the location and temporal extent of actions in the three datasets, and boosts initial accuracy on the three datasets. We additionally demonstrate the advantages of curriculum learning during this update process, and the robustness of our approach to the initial parameters of the sampling distributions. When single timestamps are consistently within the action boundaries, our approach is comparable to strongly supervised models on all datasets.

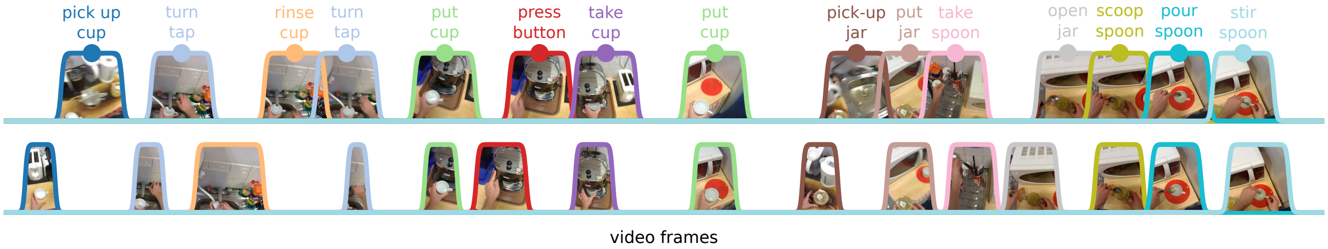

Figure 1. Replacing action boundaries with sampling distributions in an untrimmed video, given single timestamps (coloured dots at the centre of each distribution). The initial distributions (top) may overlap (e.g. 'put jar', 'take spoon') and contain background frames. We iteratively refine the distributions (bottom) using the classifier response during training.

2. Related Work

We review recent works using weak temporal labels for action recognition and localisation. For a review of works that use strong supervision, we refer the reader to [13]. We divide the section into works using video-level, transcript and point-level supervision.

Video-level supervision provides the weakest cue, signalling only the presence or absence of an action in an untrimmed video, discarding any temporal ordering. When only a few different actions are present in an untrimmed video, video-level supervision can prove sufficient to learn the actions even from long videos, as recently shown in [32, 22, 31, 28, 23]. In these works, the authors use such supervision to train a model for action classification and localisation, achieving results often comparable to those obtained with strongly supervised approaches. However, all these works evaluate their approach on the THUMOS 14 [15] and Activity Net [11] datasets, which contain mainly one class per training video. In this work, we show that as the number of different actions per training video increases, video-level labels do not offer sufficient supervision.

Transcript supervision offers an ordered list of action labels in untrimmed videos, without any temporal annotations [4, 5, 14, 24, 18, 25, 26, 10]. Some works [10, 18, 24] assume the transcript includes knowledge of 'background', specifying whether the actions occur in succession or with gaps. In [10], uniform sampling of the video is followed by iterative refinement of the action boundaries. The refinement uses the pairwise comparison of softmax scores for class labels around each boundary, along with linear interpolation. This iterative boundary refinement strategy is conceptually similar to ours. However, the approach in [10] assumes no gaps are allowed between neighbouring actions. This requires knowledge of background labels in order for the method to operate.

Point-level supervision refers to using a single pixel or a single frame as a form of supervision. This was attempted for semantic segmentation, by annotating single points in static images [2] and subsequently used for videos [20, 7]. In [20] a single pixel is used to annotate the action, in a subset of frames, both spatially and temporally. When combining this weak supervision with action proposals, the authors show that it is possible to achieve comparable results to those obtained with full and much more expensive per-frame bounding boxes. More recently, several forms of weak supervision, including single temporal points, are evaluated in [7] for the task of spatio-temporal action localisation. This work uses an off-the-shelf human detector to extract human tracks from the videos, integrating these with the various annotations in a unified framework based on discriminative clustering.

In this work, we also use a single temporal point per action for fine-grained recognition in videos. However, unlike the works above [20, 7] which consider the given annotations correct, we actively refine the temporal scope of the given supervision, under the assumption that the given annotated points may be misaligned with the actions and thus lead to incorrect supervision. We show this to effectively converge, when tested on three datasets with varying complexity, in the number of different actions in untrimmed training videos. We detail our method next.

3. Recognition from Single Timestamp Supervision

In this work, we consider the case where a set of untrimmed videos, containing multiple different actions, are provided for the task of fine-grained action recognition. That is the task of training a classifier $f(x) = y$ that takes a frame (or a set of frames) x as input to recognise a class y from the visual content of x. Our method is classifier agnostic, i.e. we do not make any assumptions about the nature of the classifier.

The typical annotation for this task is given by the actions' start and end times, which delimit the temporal scope of each action in the untrimmed video, as well as the class labels. We refer to this labelling as *temporal bounds annotation*. When using this supervision, the classifier can be trained using frames between the corresponding start/end timestamps. When replacing these annotations with a single timestamp per action instance, training a classifier is not

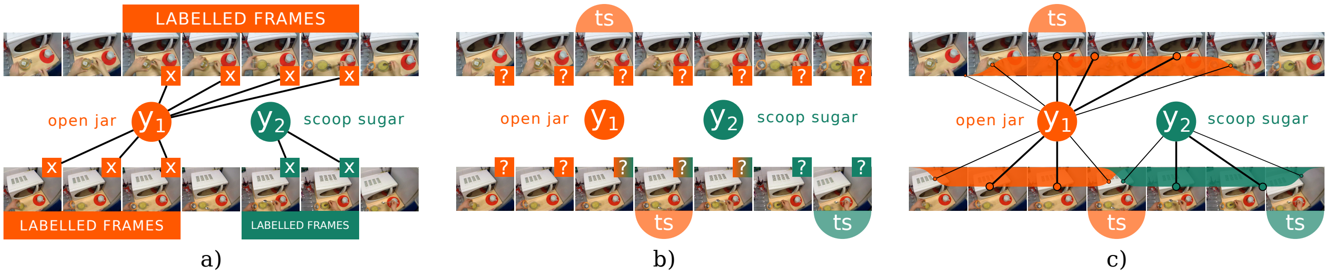

Figure 2. When start/end times are available (a), all frames within labelled boundaries can be assigned to the class label. Since action bounds are not available (b), our method aims to iteratively update the mapping between frames and class labels (c). Top and bottom plots depict different videos.

straightforward. Figure 2 compares temporal bounds (a) to the single timestamp annotations (b). In Figure 2b, it is not evident which frames could be used to train the classifier when only roughly aligned single timestamps are available. While being close to the action, the frame corresponding to the single timestamp could represent the background or another action. Additionally, the extent of the action is unknown. Our method is based on the reasonable assumption that multiple instances of each class have been labelled, allowing the model to converge to the correct frames.

We propose a sampling distribution (Section 3.1) to select training frames for a classifier starting from the annotated timestamps, as depicted in Figure 2c. After initialisation (Section 3.2), we iteratively update the parameters of the sampling distributions based on the classifier's response, in the attempt to correct misplaced timestamps and reinforce the classifier with more relevant frames (Section 3.3).

3.1. Sampling Distribution

We propose to replace the unavailable action bounds with a sampling distribution that can be used to select frames for training a classifier. For simplicity, we assume here our classifier is frame-based and takes as input a single frame. We relax this assumption later.

We argue that the sampling distribution should resemble the output of a strong classifier, i.e. a plateau of high classification scores for consecutive frames containing the action, with low response elsewhere. Another desirable property of this function is differentiability, so that it can be learnt or tuned. The Gaussian probability density function (pdf) is commonly used to model likelihoods, however it does not exhibit a plateau response, peaking instead around the mean and steadily dropping from the peak. The gate function by definition exhibits a sharp plateau, however it is not differentiable. We propose the following function to model the probability density of the sampling distributions:

$$g(x \mid c, w, s) = \frac{1}{(e^{s(x-c-w)}+1)(e^{s(-x+c-w)}+1)} \tag{1}$$

The parameter c models the centre of the plateau, while w and s model respectively its width (equal to $2w$) and the steepness of its side slopes. The range of the function is $[0,1]$. In our setting, g is defined over the frames x of an untrimmed video. We refer to g as the **plateau function** for the remainder of the text.

3.2. Initialising the Model

We initialise the sampling distributions from the single timestamp annotations. Let a_i^v be the i-*th* single timestamp in an untrimmed video v and let y_i^v be its corresponding class label, with $i \in \{1..N_v\}$ and $v \in \{1..M\}$. For each a_i^v, we initialise a sampling distribution centred on the timestamp, with default parameters w and s. We denote the parameters of the corresponding sampling distribution with $\beta_i^v = (c_i^v, w_i^v, s_i^v)$, where $c_i^v = a_i^v$, and accordingly we denote the corresponding sampling distribution with $G(\beta_i^v)$. We will use $G(\beta_i^v)$ to sample training frames for the class indicated by y_i^v.

Note that, due to the close proximity of some timestamps, the initialised plateaus may overlap considerably (Figure 1, top). We could decrease the overlap by shrinking the plateaus. However, given that we do not know the temporal extent of the actions, this may result in missing important frames. We choose to allow the overlap, and set w and s to default values that give all actions the chance to be learnt from the same number of frames.

Frames sampled from these distributions might be background frames, or be associated with incorrect action labels. To decrease noise, we rank frames sampled from all untrimmed videos based on the classifier's response, and select the most confident frames for training, inspired by curriculum learning [3]. Let $P(k|x)$ denote the softmax scores of a frame x for a class k. Let:

$$\mathcal{F}^k = \Big(x \leftarrow G(\beta_i^v) \;:\; y_i^v = k, \forall i \in \{1..N_v\}, \forall v \in \{1..M\}\Big)$$
$$s.t.\ P(k|\mathcal{F}^k_{t-1}) \geq P(k|\mathcal{F}^k_t) \tag{2}$$

be all the sampled frames from the distributions with corresponding class k, ordered according their softmax scores. We select the top T frames in $\mathcal{F}^k$ for training:

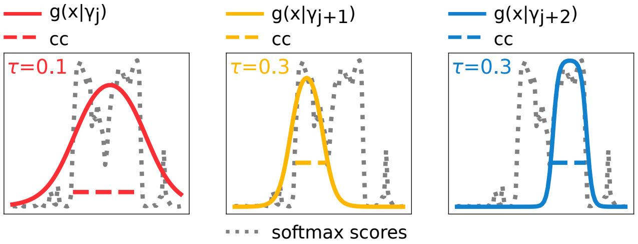

Figure 3. Finding multiple update proposals. 'cc' denotes the connected components used to fit the softmax scores.

$$\left(\mathcal{F}^k_t\right)_{t=1}^{T} \;:\; T = h|\mathcal{F}^k|,\; h \in [0,1] \tag{3}$$

With this approach, we select the frames where the classifier is most confident, which amounts to selecting the most relevant frames for each class within the plateaus. Note that Equation 3 ranks frames from all videos, and thus is independent of the number of action repetitions in one video. While with this strategy we feed the classifier fewer noisy samples, we are still potentially missing relevant frames outside the initial plateaus. After training the base model, we proceed to update the sampling distributions aiming to correct misplaced plateaus so that we can feed more relevant frames.

3.3. Updating the Distribution Parameters

We assume that, overall, the initial plateaus are reasonably aligned with the actions. Under such assumptions, we iteratively update the sampling distributions parameters, reshaping and moving the initialised plateaus over more relevant frames, in order to reinforce the classifier. We first produce update proposals from the softmax scores, then rank the proposals to select the parameters that provide the most confident updates.

Finding Update Proposals For each sampling distribution $G(\beta^v_i)$, we find update proposals given the softmax scores for the corresponding class $k = y^v_i$. For simplicity, we describe this process for one sampling distribution and the softmax scores of its corresponding class k.

We fit the pdf in Equation 1 to the softmax scores at multiple positions and temporal scales. This is done through setting a threshold $\tau \in [0,1]$ over the softmax scores, and finding all the connected components of consecutive frames with softmax scores above τ. For each connected component, we fit the pdf and consider the resulting fitted parameters as one candidate for updating the sampling distribution. As τ is varied, multiple proposals at various scales can be produced. Figure 3 illustrates an example of three update proposals, where both the position and scale of the action are ambiguous, i.e. it is unclear which plateau is the best fit.

We denote each update proposal with $\gamma^v_j = (c^v_j, w^v_j, s^v_j)$. The set of update proposals for β^v_i is thus:

$$\mathcal{Q}^v_i = \left\{\gamma^v_j \;:\; c^v_{i-1} < c^v_j < c^v_{i+1}\right\} \tag{4}$$

Note that the constraint $c^v_{i-1} < c^v_j < c^v_{i+1}$ enforces the order of the actions in v to be respected.

Selecting the Update Proposals We first define the score ρ for a given plateau function $g(x|\beta^v_i)$ by averaging the softmax scores enclosed by the plateau as follows. Let $\mathcal{X}$ be the set of frames such that $\mathcal{X} = \{x \;:\; g(x|\beta^v_i) > 0.5\}$. The score is then defined as follows:

$$\rho(\beta^v_i) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} P(y^v_i|x) \tag{5}$$

We define the confidence ψ of each proposal $\gamma^v_j \in \mathcal{Q}^v_i$ as:

$$\psi(\gamma^v_j) = \rho(\gamma^v_j) - \rho(\beta^v_i) \tag{6}$$

The underlying idea is to reward proposals whose plateaus contain frames that, on average, are scoring higher than those contained within the plateau to be updated, and thus are likely to be more relevant to the action. Accordingly, we discard update proposals with nonpositive confidence. We select the proposal $\widehat{\gamma^v_i}$ with highest confidence for each β^v_i:

$$\widehat{\gamma^v_i} = \operatorname*{arg\,max}_{\gamma^v_j} \psi(\gamma^v_j) \;:\; \gamma^v_j \in \mathcal{Q}^v_i \tag{7}$$

Updating Proposals We adopt a curriculum learning paradigm for the update as well, updating only distributions for which the selected proposals have high scores. Let:

$$\begin{aligned}\Gamma = \Big(\widehat{\gamma^v_i}\,, \forall i \in \{1..N_v\}, \forall v \in \{1..M\}\Big)& \\ s.t.\; \psi(\Gamma_{t-1}) \geq \psi(\Gamma_t)&\end{aligned} \tag{8}$$

be the sequence of all selected update proposals ordered according their confidence. We pick the top R proposals in Γ to update the corresponding sampling distributions:

$$\Gamma^R = \left(\Gamma_t\right)_{t=1}^{R} \;:\; R = z|\Gamma|,\; z \in [0,1] \tag{9}$$

The corresponding sampling distribution parameters β^v_i are then updated as follows:

$$\forall \widehat{\gamma^v_i} \in \Gamma^R \rightarrow \beta^v_i = \beta^v_i - \Lambda\Big(\beta^v_i - \widehat{\gamma^v_i}\Big) \tag{10}$$

where $\Lambda = \{\lambda_c, \lambda_w, \lambda_s\}$ denotes the set of hyperparameters controlling the velocity of the update. Note that we use a different update rate for the various parameters (c, w, s):

$$c^v_i = c^v_i - \lambda_c\Big(c^v_i - \widehat{c^v_i}\Big) \tag{11}$$

and similarly for w^v_i and s^v_i. We update proposals until convergence. This is readily assessed by observing the average confidence of the selected proposals approaching 0.

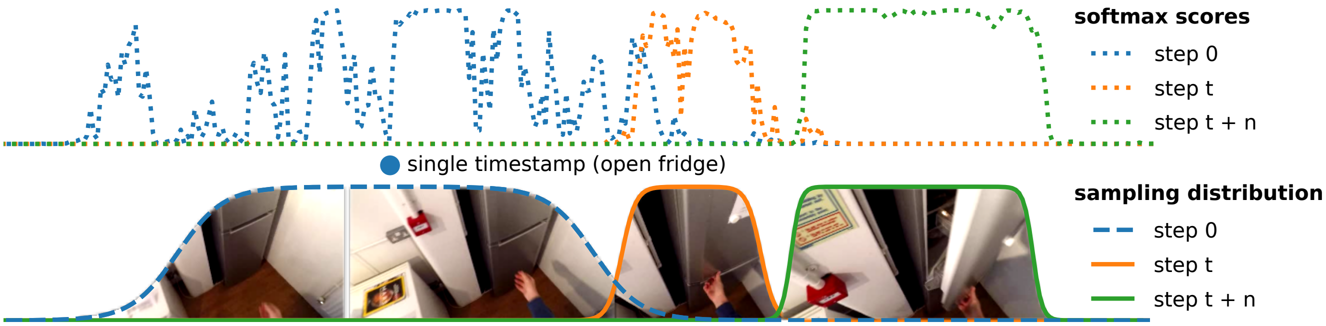

Figure 4. Updating the sampling distribution using the classifier response - example from action ‘open fridge’ in EPIC Kitchens [8]. Different colours indicate different training iterations.

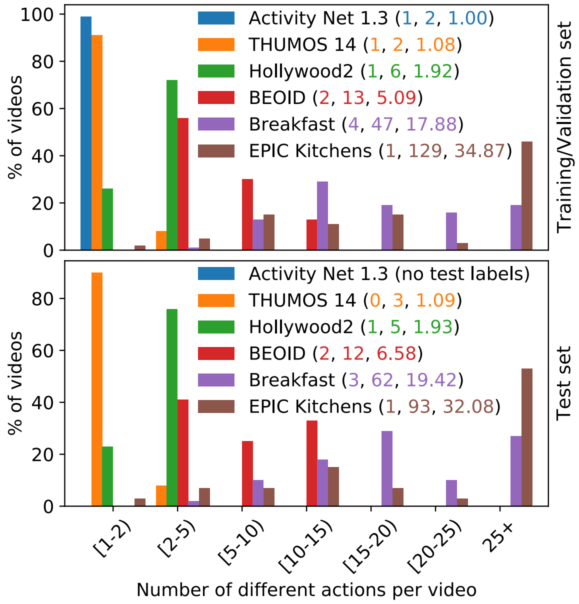

Figure 5. Different actions per video for various datasets. Numbers between parenthesis indicate (min, max, average) unique actions per video. For Activity Net [11] both train and validation sets were considered, while for THUMOS 14 [15] we considered only the validation set. We used the ‘s1’ split and fine segmentation labels for Breakfast [17]. For EPIC Kitchens [8] we consider only the videos in the used subset.

Figure 4 shows an example for updating one sampling distribution for class ‘open fridge’. The labelled timestamp and the corresponding initial sampling distribution (dotted blue and dashed blue lines) are not well aligned with the action, both positioned before the actual occurrence of the action. After a few iterations, the classifier is predicting the action with more confidence over frames located outside the initial plateau (dotted orange, top). The final sampling distribution (solid green, bottom) successfully aligns with the frames of the subject opening the fridge.

Set	Dataset	N. of classes	N. of videos	N. of actions	Avg video length	Avg classes per video	Avg actions per video
Train	THUMOS 14	20	200	3003	208.90	1.08	15.01
	BEOID	34	46	594	61.31	5.09	12.91
	EPIC Kitchens	274	79	7060	477.37	34.87	89.36
Test	THUMOS 14	20	210	3307	217.16	1.09	15.74
	BEOID	34	12	148	57.78	6.58	12.33
	EPIC Kitchens	274	26	1949	399.62	32.08	74.96

Table 1. Datasets information. Average video length is in seconds.

4. Experiments

4.1. Datasets

Figure 5 compares various common datasets [11, 15, 19, 9, 17, 8] for action recognition and localisation, based on the number of different actions per video in both train (top) and test (bottom) sets. The figure shows how these datasets range from an average of one action per video (Activity Net, THUMOS 14) to a maximum average of 34 actions per video (EPIC Kitchens). When learning from untrimmed videos with weak temporal supervision, the number of *different* actions per video plays a crucial role. We thus evaluate our method covering this spectrum by selecting three datasets with increasing number of classes per video, namely THUMOS 14 [15], BEOID [9] and EPIC Kitchens [8]. We show in Section 4.4 that, as the number of different actions per video increases, video-level labels no longer provide sufficient temporal supervision, while single timestamps constitute a valid compromise between annotation effort and accuracy.

For THUMOS 14 we use the subset of videos that were temporally labelled for 20 classes, while for BEOID we randomly split the untrimmed videos in an 80-20% proportion for training and testing. For EPIC Kitchens we use a subset of the dataset selecting participants P03, P04, P08 and P22. With a total of 13.5 hours footage length this subset amounts to 25% of full the dataset. Table 1 summarises various statistics of the chosen datasets. Despite considering a subset of the full dataset, EPIC Kitchens is by far the most challenging, given its very long videos containing many different actions. Additionally, EPIC-Kitchens offers novel narration annotations, as we discuss in Section 4.3.

4.2. Implementation Details

We use the Inception architecture with Batch Normalisation (BN-Inception) [27] pre-trained on Kinetics [6], and use TV-L1 optical flow images [35], with stack size 5. For training, we sample 5 stacks per action instance, and use average consensus as proposed in [33]. When comparing to full temporal supervision using the start/end action times, the stacks are sampled randomly within equally sized snippets, as in [33]. For faster evaluation, we uniformly sample 10 stacks from the trimmed test videos and take the centre crop using the average score for the final prediction. We use Adam Optimiser with batch size 256, fixed learning rate equal to 10^{-4}, dropout equal to 0.7 and no weight decay.

We initialise the sampling distributions with $w = 45$ frames (1.5 seconds at 30 fps) and $s = 0.75$ for all datasets. As we show in Section 4.4, our method is robust to the choice of the initial parameters. We train the base model for 500 epochs, to ensure a sufficient initialisation, then update the sampling distributions running the method for 500 additional epochs. The initial 500 epochs were largely sufficient for the test error to converge in all experiments before the update started. After training the base model with curriculum learning, we gradually increase h (see Equation 3) until reaching $h = 1$, which corresponds to using all the sampled frames. We use a fixed $z = 0.25$ to select the top R update proposals (see Equation 9). We vary h to control noise in training frames, and keep z fixed. Increasing z primarily speeds the update of the distribution parameters and is akin to changing the method's learning rate. To produce the update proposals, we use $\tau \in \{0.1, 0.2, \ldots, 1\}$ and discard connected components shorter than 15 frames. We set update parameters $(\lambda_c, \lambda_w, \lambda_s) = (0.5, 0.25, 0.25)$ for all datasets, updating the sampling distributions every 20 epochs. Our code uses PyTorch and is publicly available[1].

4.3. Single Timestamps

The EPIC Kitchens dataset [8] was annotated using a two stages approach: videos were firstly narrated by the participants, through audio live narration, to produce a rough temporal location of the performed actions, from which action boundaries were then refined using crowd sourcing. We use the **narration** start timestamp as our single timestamp for training. These timestamps come from the narration audio track and exhibit a challenging offset with respect to the actions occurrences in the videos: 55.8% of the narration timestamps are not contained in the corresponding labelled boundaries. For the timestamps outside the bounds, the maximum, average and standard deviation distance to the labelled boundaries were respectively 11.2, 1.4 and 1.6 seconds. To the best of our knowledge, this paper offers the first attempt to train for fine-grained action recognition on EPIC Kitchens using only the narration timestamps.

Dataset	**CL *h***	**Before update**	**After update**
THUMOS 14	0.25	26.10	28.88
	0.50	32.69	55.15
	0.75	33.59	56.42
	1.00	63.41	63.53
BEOID	0.25	47.97	52.70
	0.50	71.62	83.11
	0.75	74.32	83.11
	1.00	64.86	70.27
EPIC Kitchens	0.25	20.47	22.83
	0.50	21.39	25.35
	0.75	20.73	23.86
	1.00	23.55	24.17

Table 2. Top-1 accuracy obtained with single timestamp supervision on the **TS** point set. CL h indicates the h parameter used for training the base model (see Equation 3).

THUMOS 14 and BEOID do not have single timestamp annotations. We simulate *rough* single timestamps from the available labels, drawing each a_i from the *uniform* distribution $[\sigma_i - 1sec, \epsilon_i + 1sec]$, where σ_i and ϵ_i denote the labelled start and end times of action i. This approximately simulates the same live commentary annotation approach of EPIC Kitchens. We refer to this set of annotations as **TS**.

We also use another set of single timestamps for all the three datasets, where each a_i is sampled using a normal distribution with mean $\frac{\sigma_i + \epsilon_i}{2}$ and standard deviation $1sec$. This assumes that annotators are likely to select a point close to the middle of the action when asked to provide only one timestamp. We refer to this second set of points as **TS in GT**.

4.4. Results

The evaluation metric used for all experiments is top-1 accuracy. We first evaluate the **TS** timestamps with curriculum learning (CL) for training the base model running experiments with $h \in \{0.25, 0.50, 0.75\}$, as well as using all the sampled frames for training ($h = 1$).

As shown in Table 2, results obtained after the update consistently outperform those obtained before the update, for all datasets and for all h values. For BEOID and EPIC, our CL strategy reduces the amount of noisy frames when training the base model, i.e. the best results are obtained with $h = 0.50$. However, on THUMOS 14, the CL approach for the base model is less effective, with the best performance achieved when all frames are used in training. We further analyse this in Figure 7, which illustrates

[1] https://bitbucket.org/dmoltisanti/action_recognition_single_timestamps/

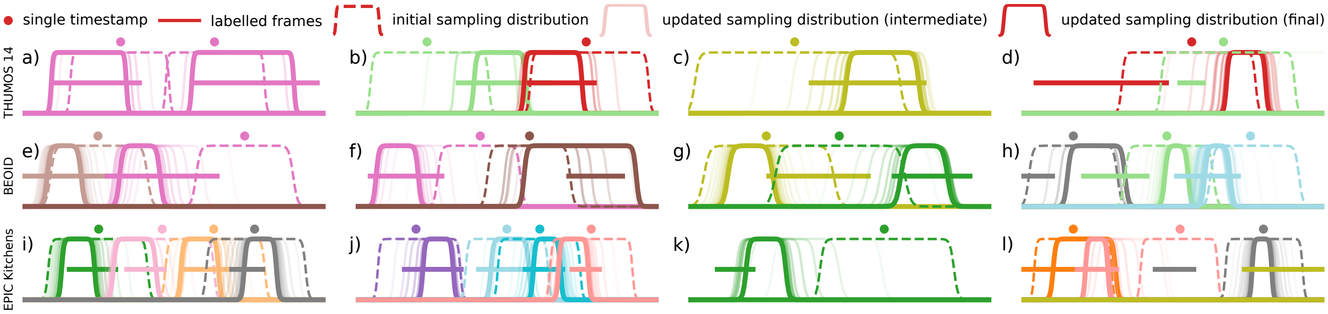

Figure 6. Qualitative results on the three datasets, plotted from results obtained with CL $h = 0.50$. Different colours indicate different classes on a per-dataset basis. Labelled frames used only for plotting. Video with class labels in supplementary material.

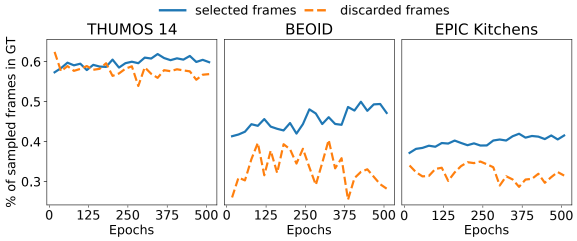

Figure 7. Percentage of sampled framed contained within labelled bounds, over training epochs (CL $h = 0.50$, before update).

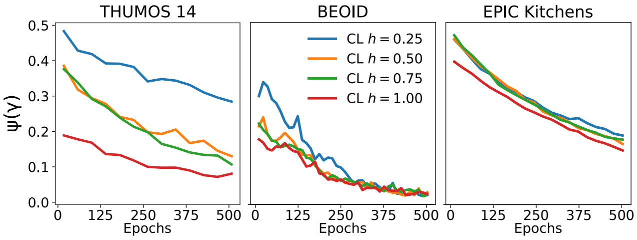

Figure 8. Average confidence of selected update proposals, as calculated in Equation 6, over training epochs.

the percentage of selected and discarded frames that were enclosed by the labelled action boundaries (used only for plotting), before update. For BEOID and EPIC Kitchens, we notice a neat separation between the selected and discarded frames. This shows that the CL strategy was effectively picking the most relevant frames within the plateaus during training. For THUMOS 14, we do not observe the same distinct trend. A balance between the plateau width and the number of sampled frames might resolve this, but we leave this for future work.

In Figure 8, we assess the update convergence by plotting the average confidence of the selected update proposals over training epochs. For all cases, the average confidence decreases steadily, indicating the classifier's convergence.

We illustrate a few examples from each dataset in Figure 6, showing the iterative update of the sampling distributions. The examples are plotted from results obtained with CL $h = 0.50$ on the **TS** point set. Our update method is able to successfully refine the sampling distributions even when the initial plateaus are considerably overlapping with other unrelated actions (subplots *e, g, i, j*) or when the initial plateaus contain much background (subplots *b, c, e, f, k*). We highlight a few failure cases as well. In subplots *g* (light green plateau) and *h* (grey plateau), the initial plateaus are pushed outside the relevant frames. In both cases, the number of training examples was small (8 and 5 instances), with the single timestamps located almost always outside the action. In subplot *l*, the pink and grey initial plateaus were shifted with respect to the corresponding actions, reflecting the challenge EPIC Kitchens poses when using narration timestamps. While the update method managed to recover the correct location for the pink plateau, the grey plateau did not converge to the relevant frames.

Parameters initialisation We assess the impact of the initial parameters w and s for the sampling distributions through a grid search. Figure 9 compares top-1 accuracy obtained after update with different (w, s) combinations, using CL $h = 1.00$. We observe that for the two large datasets (THUMOS 14 and EPIC Kitchens), our method is robust to the initialisation of both w and s, i.e. similar performance is obtained for all parameters combinations. Decreased robustness for BEOID is potentially due to the small size of the dataset.

We note that the best results obtained via the grid search (highlighted with red boxes in the Figure) are slightly superior to those previously reported in Table 2. This is because when the plateaus are optimally initialised, we are less likely to sample noisy frames when training.

4.5. Comparing Levels of Supervision

We now compare different levels of temporal supervision, namely the weakest video-level labels, single timestamps (both **TS** and **TS in GT** point sets) and full temporal boundaries. Particularly, we show that video-level supervision, while being the least expensive to gather, cannot provide a sufficient supervision when dealing with videos containing multiple different actions.

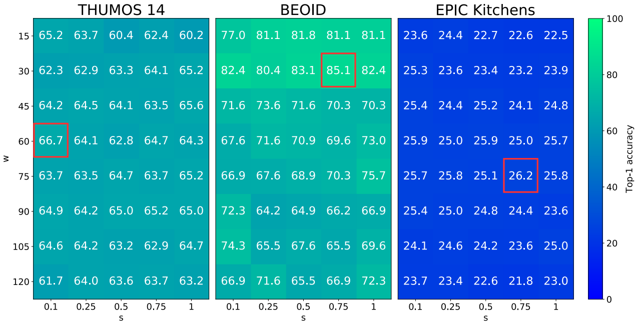

Figure 9. Top-1 accuracy obtained after update with different initial w and s, with CL $h = 1$. Red boxes highlight best results.

We choose Untrimmed Net [32] amongst the aforementioned works [32, 22, 28, 23], which is used to extract features in [23] and is the backbone model of [28], due to the availability of published code. We train Untrimmed Net using uniform sampling and hard selection module, using the same BN-Inception architecture and Kinetics pre-trained weights used for our baselines. For Untrimmed Net we report results obtained on RGB images as these performed better than flow images in all our experiments.

Table 3 compares the results obtained with the three temporal supervisions. Single timestamps results are reported after update, with CL $h = 1.00$. When only one class of action is contained in the videos, as in THUMOS 14, Untrimmed Net notably achieves virtually the same results as the fully supervised baseline. However, as the average number of different actions per video increases, it becomes increasingly harder for video-level supervision to achieve sufficient accuracy. In [32] when a video contains action instances from multiple classes, the label vector is L^1-normalised so that all the present classes contribute equally to the cross-entropy loss. As a consequence, without any temporal labels, it is very hard to train the model when a large number of classes are present in a video.

Results obtained with single timestamps remain comparable to full supervision for all datasets, though requiring significantly less labelling effort[2]. For THUMOS 14 and BEOID, we observe little difference between the point sets **TS** and **TS in GT**. For EPIC Kitchens, which has the largest number of distinct classes per video, we notice a larger gap in performance with respect to the fully supervised baseline. However, when drawing the initial timestamps from the labelled bounds (**TS in GT**), we achieve higher accuracy. From these results we conclude that single timestamps supervision constitutes a good compromise between accuracy and annotation effort.

Baseline		U. Net[32]		Ours	
Supervision	**APV**	Video-level	TS	TS in GT	Full
THUMOS 14	1.08	64.92	66.68	64.53	67.10
BEOID	5.09	28.37	85.14	88.51	87.83
EPIC Kitchens	34.87	2.20	26.22	32.53	35.97

Table 3. Comparison between different levels of temporal supervision. APV indicates the average number of unique actions per training video. TS results refer to the accuracy obtained with the best initialisation (see Figure 9). Timestamp results are reported after update, with $h = 1.00$.

Baseline	**mAP@0.1**	**mAP@0.2**	**mAP@0.3**	**mAP@0.4**	**mAP@0.5**
Ours (Full)	26.7	22.5	18.5	14.3	11.1
Ours (TS)	24.3	19.9	15.9	12.5	9.0
U. Net [32]	44.4	37.7	28.2	21.1	13.7

Table 4. Localisation results on THUMOS 14 at different IoUs.

4.6. Future Direction: Localisation with TS

In this work we focus on single timestamp supervision for action classification. Using solely frame-level classification scores to localise the extent of actions would be suboptimal. We show this in Table 4, which presents mean average precision (mAP) on THUMOS 14 obtained with our baselines, compared to [32]. We follow the localisation pipeline of [32], fusing RGB and flow scores obtained with full and single timestamp (TS) supervision. While TS performs comparably to full supervision, even full supervision is inferior to [32], which is optimised for localisation. Our approach could be extended to localisation by supervising a temporal model (e.g. RNN) from plateau functions to learn the temporal boundaries. We leave this for future work.

5. Conclusions

In this work we investigate using single timestamp supervision for training multi-class action recognition from untrimmed videos. We propose a method that initialises and iteratively updates sampling distributions to select relevant training frames, using the classifier's response. We test our approach on three datasets, with increasing number of unique action classes in training videos. We show that, compared to video-level supervision, our method is able to converge to the locations and extents of action instances, using only single timestamp supervision. Results also demonstrate that, despite using a much less burdensome annotation effort, we are able to achieve comparable results to those obtained with full, expensive, temporal supervision. Extending these annotations to other tasks such as localisation is left for future work. Future directions also include updating the sampling distribution parameters in an end-to-end differentiable manner.

Acknowledgement Research supported by EPSRC LOCATE (EP/N033779/1) and EPSRC DTP. We use publicly available datasets, and publish our code.

[2] For completion, accuracy before update for TS was 64.74, 73.65 and 25.19 for THUMOS 14, BEOID and EPIC Kitchens. For TS in GT, accuracy before update was 64.74, 85.81 and 31.66.

References

[1] Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 1

[2] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016. 1, 2

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, 2009. 3

[4] Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 2

[5] Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 2

[6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1, 6

[7] Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid. A flexible model for training action localization with varying levels of supervision. *arXiv preprint arXiv:1806.11328*, 2018. 1, 2

[8] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The EPIC-KITCHENS Dataset. In *ECCV*, 2018. 1, 5, 6

[9] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. You-do, I-learn: Discovering task relevant objects and their modes of interaction from multi-user egocentric video. In *BMVC*, 2014. 1, 5

[10] Li Ding and Chenliang Xu. Weakly-supervised action segmentation with iterative soft boundary assignment. In *CVPR*, 2018. 2

[11] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2, 5

[12] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 1

[13] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. In *Image and vision computing*, 2017. 2

[14] DeAn Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016. 2

[15] Yu-Gang Jiang, Jingen Liu, Amir R. Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`, 2014. 1, 2, 5

[16] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017. 1

[17] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 5

[18] Hilde Kuehne, Alexander Richard, and Juergen Gall. Weakly supervised learning of actions from transcripts. In *CVIU*, 2017. 2

[19] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *CVPR*, 2009. 5

[20] Pascal Mettes, Jan C. van Gemert, and Cees G.M. Snoek. Spot on: Action localization from pointly-supervised proposals. In *ECCV*, 2016. 2

[21] Davide Moltisanti, Michael Wray, Walterio Mayol-Cuevas, and Dima Damen. Trespassing the boundaries: Labeling temporal bounds for object interactions in egocentric video. In *ICCV*, 2017. 1

[22] Phuc Nguyen, Ting Liu, Gautam Prasad, and Bohyung Han. Weakly supervised action localization by sparse temporal pooling network. In *CVPR*, 2018. 2, 8

[23] Sujoy Paul, Sourya Roy, and Amit K Roy-Chowdhury. W-TALC: Weakly-supervised temporal activity localization and classification. In *ECCV*, 2018. 2, 8

[24] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *CVPR*, 2016. 2

[25] Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with RNN based fine-to-coarse modeling. In *CVPR*, 2017. 2

[26] Alexander Richard, Hilde Kuehne, Ahsan Iqbal, and Juergen Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. In *CVPR*, 2018. 2

[27] Ioffe Sergey and Szegedy Christian. Accelerating deep network training by reducing internal covariate shift. *CoRR*, 2015. 6

[28] Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly supervised temporal action localization in untrimmed videos. In *ECCV*, 2018. 2, 8

[29] Gunnar A. Sigurdsson, Olga Russakovsky, and Abhinav Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017. 1

[30] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1

[31] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 2

[32] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. UntrimmedNets for weakly supervised action recognition and detection. In *CVPR*, 2017. 2, 8

[33] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *ECCV*, 2016. 1, 6

[34] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 1

[35] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007. 6