Hybrid Active Learning via Deep Clustering for Video Action Detection (Supplementary Material)

Aayush J Rana
aayushjr@knights.ucf.edu

Yogesh S Rawat
yogesh@crcv.ucf.edu

Center for Research in Computer Vision (CRCV)
University of Central Florida

1. Overview

This is supplementary material for the experiments in main paper. We present detailed results for different experiments from the main paper and expand on relevant topics. We analyse the effect of sample selection using clustering approach and compare it with other baseline sample selection methods. Then we do the annotation cost to performance analysis, demonstrating the viability of low-cost sample selection using our approach. We evaluate our proposed *STeW-Loss* against other baseline loss functions in detail. Finally we expand on all technical details (network, cluster, implementation) in depth.

2. Analysis on AVA dataset

We provide results for baseline experiment on AVA using proposed training setup in table 1. We use YOWO [7] with resnet-18 (R-18) and ShuffleNetV2 (SN-V2) backbone and train on a subset of 5% random annotation and increase to [10%, 15%] annotation sampled randomly and using proposed method. Based on our results, we have three important points to note, 1) AL requires a good base model for effective sample selection, but AVA is a challenging dataset which makes sample selection difficult for AL with fewer annotations, 2) AVA has atomic actions, and average length is around 1-2 seconds, which leaves no room for intra-sample selection as only 1 keyframe is annotated per second. Videos are long, but scene changes frequently with less temporal coherency as average length of each atomic action is only **1.98 s** with only 10% instances being above 6s length, in contrast to UCF-101-24 and JHMDB-21. 3) AVA does not solve dense spatio-temporal detection as only a key-frame is annotated and all existing models predict detection on single frame.

3. Effect of number of clusters

Our objective with cluster based sample grouping is to get a general representation of the sample which is not strictly based on the class label. This makes cluster assignment easier than having to identify 24 clusters for 24 classes in UCF-101-24 (21 in case of J-HMDB-21). Our experiments in the main manuscripts are done with $K = 5$ cluster centers, but we also perform the entire AL cycle using $K = 10$ and $K = 15$ cluster centers. As shown in Figure 1, the overall performance is more or less close to each other for all values of K, while the performance for $K = 5$ is slightly better. Allowing the clusters to focus on features not tied to the classes and having fewer of such clusters performs better than having large cluster centers.

Method	Annot %	R-18	SN-v2
Random	5%	7.13	7.18
Random	10%	8.48	8.64
Ours	10%	8.61	8.81
Random	15%	9.73	9.60
Ours	15%	9.84	9.85

Table 1. Comparison on AVA using different annotation percent. We evaluate the performance using randomly selected annotations and annotations selected using our *CLAUS* method with two different backbones, Resnet-18 (R-18) and ShuffleNet-v2 (SN-v2). We report the scores for f-mAP @ 0.5 mIoU.

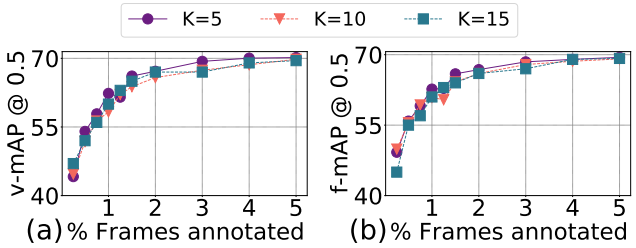

Figure 1. Comparison of model performance using different cluster centers (K). We train our proposed CLAUS using different cluster centers for UCF-101-24. We observe that model performance remains similar for different cluster center numbers.

4. Annotation selection methods

4.1. Hybrid sample selection

We evaluate the effect of sample selection for active learning using the proposed hybrid approach in table 3 and 4. Since we don't know the sample label beforehand, selecting based on uncertainty or entropy based score alone does not warrant diverse sample selection. Since it only selects based on the scores, it can cause selection of samples from a particular set of classes which are harder for the model to learn. This causes a class bias which leads the model to only focus on those classes in further active learning steps and prevents adding diverse samples for labeling. To prevent this, we use the $CLAUS$ approach, which relies on the cluster assignment of each sample to enforce selection from different clusters. This takes the uncertainty based score and cluster assignment both into account for the sample selection. The selection process is further detailed in Algorithm 1.

We evaluate the proposed $CLAUS$ method and compare with sample selection without cluster for UCF-101-24 and J-HMDB-21 in table 3 and 4 respectively, where the performance using $CLAUS$ is consistently better than non-clustering approach. The non-clustering approach uses same uncertainty based scoring method as $CLAUS$ and *STeW-Loss*.

4.2. Active learning baseline methods

To further evaluate the effect and validity of the proposed $CLAUS$ method, we compare with other active learning baseline methods for sample and frame selection based on entropy, uncertainty and random selection. All selection methods are trained using our proposed *STeW-Loss* with same training parameters (epochs, learning rate) with the results shown in table 5. Since other methods are not designed with sample selection in consideration, they often select samples which the model finds harder regardless of their similarity. These methods don't have a mechanism to guess the similarity of different samples, which limits the variation it can get during selection step. In contrast, random selection could add more variation since it does a near-uniform sampling from the given set. The proposed $CLAUS$ method uses cluster representation to check the similarity of samples and can enforce the algorithm to select samples that are less similar in representation. We see in table 5 that our $CLAUS$ method outperforms all other baseline methods for sample and frame selection. We also show the comparison of per class sample selection in Figure 2 for all baselines and show how our method selects small set of frames for annotation in Figure 3.

4.3. Error margins

We provide scores for average of 3 runs for each variation. We noticed that random and equidistant selection has higher average error of ± 1.03 and 0.84 respectively, and AL based selection has lower error with *CLAUS*=± 0.20, entropy=± 0.19 uncertainty=± 0.35.

Selection Strategy	UCF-101-24		J-HMDB-21	
	v-mAP	f-mAP	v-mAP	f-mAP
Sample	36.2	42.6	53.1	56.2
Intra-sample	71.8	70.9	70.4	74.1
Hybrid (our)	72.2	72.1	71.5	72.8

Table 2. Comparison of different selection strategies. We compare the v-mAP and f-mAP scores @ 0.5 mIoU. We report at 5% annotations for UCF-101-24 and at 5.4% annotations for J-HMDB-21.

4.4. Selection strategy evaluation

We analyze the effectiveness of different selection strategy included in the main paper: sample selection, intra-sample selection and hybrid selection. As shown in table 2, sample selection performs worst as the budget is used up to annotate entire samples. Intra-sample selection comparatively performs better as more diverse samples are selected and our proposed hybrid selection strategy performs better than both as it selects diverse frames with less budget on less relevant samples.

5. Budget utilization

5.1. Cost-performance analysis

We gradually increase annotations based on a fixed cost and evaluate them to find at which point is the annotation enough to get comparable performance with fully-supervised methods. For each active learning step s, we assume a constant budget of B_v^s, B_f^s for sample and frame annotation, which limits the total annotation cost for that step to be $Cost = C_v^s + C_f^s$. For UCF-101-24 we fix that cost to 1000 per round, which is divided into selecting 5% of samples to annotate in each round where we also annotate 5% of frames from those samples. Similarly, we fix the cost of J-HMDB-21 to 34 per round. We demonstrate that the overall annotation cost-performance has a linear trend for both datasets in table 6 and 7, with UCF-101-24 getting comparable results with 5% of total frames annotated with annotation cost of $16,000$ (vs $403,282$ for fully-supervised). Similarly, J-HMDB-21 has comparable results with 5.4% annotations with annotation cost of $1,226$ (vs $22,712$ for fully-supervised). We also demonstrate the cost to performance graph for our and random sampling on UCF-101-24 in Figure 4.

5.2. Sample+frame vs only frame increment

Annotation increment in each active learning step can be done in multiple ways, one being increasing both samples and frames annotation together as shown in table 6 and 7.

Algorithm 1 Iterative video and frame selection algorithm

Input: videos $\mathcal{V}_L, \mathcal{V}_U$, frames $\mathcal{F}_L, \mathcal{F}_U$, budget $\mathcal{B}_v, \mathcal{B}_f$
Initialize: total cost c_v^{total}, c_a^{total} = 0, $V_{annot} \leftarrow \{\}$, $F_{annot} \leftarrow \{\}$, cluster $\mathcal{C}$ with K centers, sample count per cluster $\mathcal{C}_s = [0, 0, ..0]_K$, $V\_score \leftarrow \{\}$

for all videos v in $\mathcal{V}_U$ **do**
  $U_{scores} \leftarrow \{\}$
  for all frame f in v **do**
    U^f = $[\sum_{i=0}^{T} P(f, i)]$ / T *// Get frame's uncertainty over T runs using Eq. 1*
    Append U^f to U_{scores}
  end for
  for all A_t **do**
    $index$ = $max(U_{scores})$
    $U^{sorted} \leftarrow index$
    $U_{scores}^{i-W:i+w}$ = Distance(U^{index}, $i - w : i + w$) $\times$ $U^{i-w:i+w}$
    // Update nearby frame's score based on distance to current selection, within a window w
  end for
  Append $[v_s \leftarrow \sum U^{sorted}]$ to V_score *// Get video score from top A_t frames*
end for
while $c_v^{total} \leq B_v$ or $c_a^{total} \leq B_a$ **do**
  // Loop until cost<budget
  $v = max(V\_score)$ *// Get video with highest uncertainty score*
  $\mathcal{C}_V$ = Cluster(v) *// Get video's cluster assignment using trained model*
  if $\mathcal{C}_s[\mathcal{C}_V] \leq$ cluster limit **then**
    // Check if video from saturated cluster
    Append v to V_{annot}, $V\_score[v] = 0$ *// Mark v for annotation and reset score*
    $[c_v^{total}, \mathcal{C}_s[\mathcal{C}(V)]] += 1$ *// Update video annotation cost and cluster count*
    Select $\mathcal{F}_A$ frames using U^{sorted} *// Select $A_t\%$ frames from a sorted list as above*
    Append $\mathcal{F}_A$ to F_{annot}, $c_a^{total} += 1$
  end if
end while
return (V_{annot},F_{annot})
=0

%Frame	video-mAP@				frame-mAP@			
Annot	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
With Cluster								
0.25	94.91	87.98	78.34	45.00	86.59	82.03	74.92	50.11
0.50	97.24	92.12	85.00	54.34	90.84	86.87	80.71	56.59
0.75	97.65	93.63	88.25	57.68	91.61	88.70	83.12	59.49
1.00	98.12	95.85	88.91	61.84	93.62	90.91	85.80	61.63
1.25	98.25	95.95	89.97	65.55	93.42	90.94	86.44	65.61
Without Cluster								
0.25	92.15	84.96	75.60	45.00	80.43	76.35	70.28	45.00
0.50	96.25	91.04	83.96	51.24	88.78	85.25	78.69	53.10
0.75	97.96	92.89	84.16	55.23	90.78	87.69	79.36	56.03
1.00	98.04	95.35	84.57	59.43	93.18	90.39	80.46	59.66
1.25	98.18	95.49	87.28	61.50	93.33	90.85	84.02	62.04

Table 3. Comparing our approach with and without clustering on **UCF-101-24**.

%Frame Annot	video-mAP@ 0.1	0.2	0.3	0.5	frame-mAP@ 0.1	0.2	0.3	0.5
With Cluster								
0.15	92.41	84.01	63.81	4.69	67.97	61.96	54.12	27.19
0.30	99.60	96.71	88.17	41.61	97.08	94.01	85.34	45.31
0.45	99.80	96.75	91.00	52.54	97.22	94.77	88.53	54.75
0.60	99.85	96.75	91.36	56.00	97.09	94.30	89.36	60.47
0.75	99.83	96.82	91.33	57.61	97.06	94.36	89.19	60.93
Without Cluster								
0.15	92.41	84.01	63.81	4.69	67.97	61.96	54.12	27.19
0.30	98.05	94.17	87.49	40.07	93.27	89.07	81.51	44.39
0.45	99.21	96.25	86.06	46.52	96.34	93.65	84.63	47.91
0.60	99.20	96.17	88.17	52.48	96.80	94.04	86.54	49.86
0.75	99.21	97.15	90.26	54.03	97.02	94.20	87.27	52.10

Table 4. Comparing our approach with and without clustering on **J-HMDB-21**.

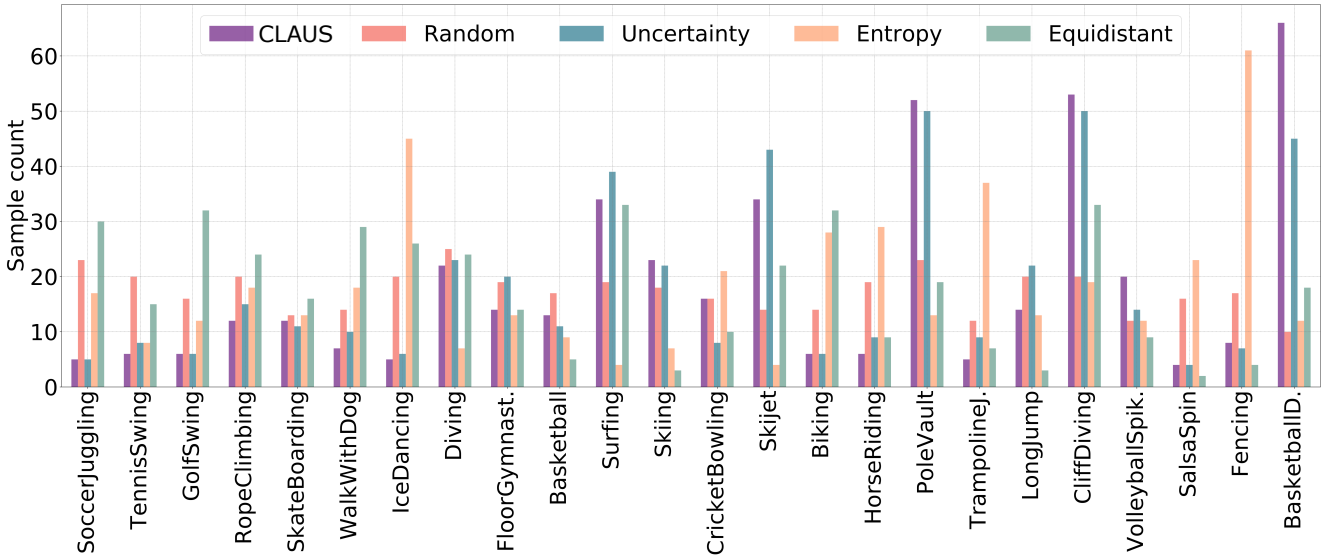

Figure 2. Comparison of per class sample count for all baseline methods on UCF-101-24 at 1% annotation. We demonstrate the per class samples selected using our method (*CLAUS*) and compare with baselines (random, equidistant, uncertainty [5], entropy [1]). We observe that while random and equidistant has almost-uniform sample selection, they have lower performance due to redundant sample selection for classes which are already doing good. All active learning baselines instead select samples based on utility, as a result have lower samples for some classes and higher for some. Our method specifically has cluster based inter-sample scoring method, which creates diversity in sample selection while selecting more from samples performing poorly.

Another option is to not select new samples, but only increase frames for existing set of samples in the given step. This option will increase more frames while not adding variation via new samples in the training set. We evaluate the effect of both variation in table 8. We increase both sample and frame in **sample+frame** variation for all the steps. For **only frame** variation, we increase only frames from 0.5% to 1.5% frame annotation step. From the table, we can observe that the **sample+frame** variation where we increase both samples and frames annotation together for a given cost gives better performance. Given our loss function which can interpolate and handle pseudo-labels during training, the model benefits from having more videos to increase training variation rather than having more frames for limited videos.

6. Loss function variations

We introduce the *Spatio-temporal weighted (STeW) Loss* to handle the interpolated pseudo-labels during training and

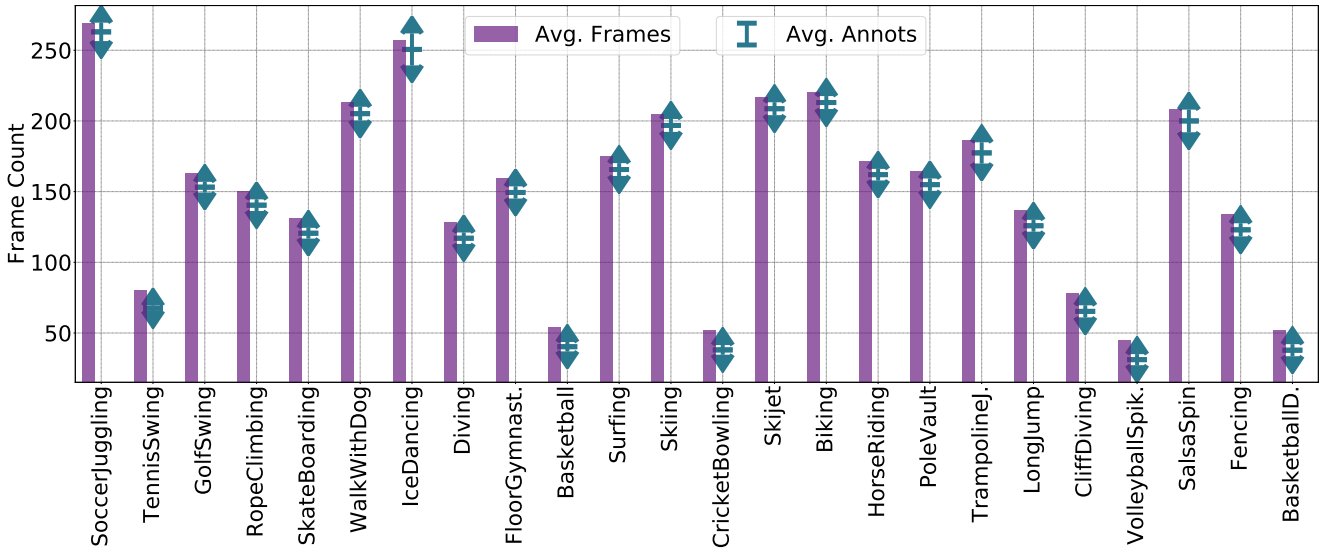

Figure 3. We show the per class average frame count and the per class average annotated frames using our proposed $CLAUS$ method on UCF-101-24 at 1% frames annotated. For each class, we show the average frames from all selected samples and compare with the average annotated frames selected using our method. We select a minor set of frames for annotation as shown by the average annotated frames while performing better than other baselines.

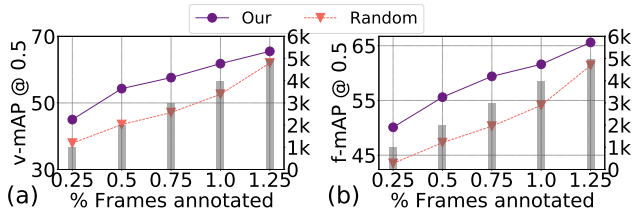

Figure 4. Performance evaluation of our method with random selection baseline on UCF-101-24 for various sample annotation percent. The cost of annotation for each step is shown by the shaded bars, with the cost value in the right axis in thousands.

adjust their impact in loss computation based on their perceived correctness, shown visually in Figure 5. We assume that pseudo-labels interpolated closer to an actual ground truth will be more likely to be correct while the ones further away might have some degree of correctness, which we model using a Gaussian distribution centered at the closest ground-truth frame. We could train without the Gaussian weight for the pseudo-labels, which would be harsher for wrongly-interpolated annotations, specially during initial stage with fewer frames annotated. A basic variation is to simply only use the available ground-truth for loss computation (Frame method) without having any interpolation. We consider all three variations and train the model with our active learning algorithm using different loss functions. The result in table 10 shows that *interpolation* performs decently while *frame* only method performs poorest. Compared to both, the proposed *STeW-Loss* outperforms them and gives best score for same cost.

6.1. Static vs Moving actions

A key concern is the ability to use pseudo-labels correctly accounting for the motion of the action. As shown in Figure 5, different actions have different speed and camera motion, which might make interpolation based pseudo-label generation non-trivial and add a lot of training noise. The proposed STeW loss gives weight per pixel for each pseudo-label based on the overlap of foreground and background in nearby frames. We see in Figure 5 that for extreme non-static motion (row 3 long jump) the actor only has small area with high weight while most actor neighboring region is almost zero weight. We do an evaluation by separating UCF-101-24 manually into static and non-static classes based on background and camera motion (excluding ambiguous classes) in table 9. The delta here shows that both static and non-static benefit from STeW loss. While other methods such as using small spatial neighborhood and camera motion based estimation are also valid ways to handle motion component in pseudo-labels, the proposed *STeW* provides an automated and less computational method than motion estimation as we only check consistency of binary action mask to get the weights.

%Frame	video-mAP@				frame-mAP@			
Annot	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
CLAUS								
0.25	94.91	87.98	78.34	45.00	86.59	82.03	74.92	50.11
0.50	97.24	92.12	85.00	54.34	90.84	86.87	80.71	56.59
0.75	97.65	93.63	88.25	57.68	91.61	88.70	83.12	59.49
1.00	98.12	95.85	88.91	61.84	93.62	90.91	85.80	61.63
1.25	98.25	95.95	89.97	65.55	93.42	90.94	86.44	65.61
Entropy								
0.25	91.25	84.87	75.18	39.06	83.98	78.06	70.28	44.53
0.50	92.15	84.99	75.59	45.62	80.43	76.12	70.23	50.43
0.75	92.56	85.88	78.78	50.64	82.43	78.33	73.06	53.77
1.00	95.69	90.14	84.63	58.22	87.96	85.99	79.59	58.25
1.25	98.15	95.03	88.60	61.10	92.79	90.14	85.16	62.36
Uncertainty								
0.25	91.65	84.69	75.37	39.00	84.25	78.36	70.26	45.12
0.50	96.82	91.48	83.24	52.81	88.96	85.11	79.32	53.74
0.75	97.68	93.58	87.55	54.61	90.87	87.12	79.65	55.84
1.00	98.01	95.37	89.55	60.27	92.88	90.17	85.50	60.29
1.25	98.14	95.58	89.62	62.11	92.97	90.55	86.21	62.18
Equidistant								
0.25	91.28	83.97	75.25	38.60	84.41	78.11	69.89	44.02
0.50	97.07	90.21	80.04	44.82	92.15	86.74	77.43	48.67
0.75	97.33	93.20	82.42	49.14	92.07	87.65	81.21	52.00
1.00	97.82	94.17	84.54	53.47	92.74	88.68	82.34	55.40
1.25	97.90	94.89	88.11	62.93	93.62	89.74	84.02	63.02
Random								
0.25	92.52	83.84	74.31	37.93	84.09	77.55	69.23	43.53
0.50	97.18	89.38	78.28	43.56	91.41	85.99	76.21	47.36
0.75	97.18	92.37	81.22	47.14	92.80	88.65	79.88	50.33
1.00	96.42	92.27	84.21	52.66	91.75	88.31	81.24	54.18
1.25	97.46	94.02	87.09	62.04	91.66	88.58	83.37	61.47

Table 5. Evaluating performance of different baseline selection methods for **UCF-101-24**. We report the performance of different cost function to select the videos and frames for annotation in each active learning step.

7. Implementation Details

7.1. Network details

We follow the implementation by [4] with some upgrades to perform video action detection. We use the Inception 3D network by [3] to extract features from the input video with T frames of $H \times W$ size. We get the features from $Mixed\_4f$ layer from the encoder (I3D) network and pass it to the 2D capsule layers. We use two capsule layers following [4] and change it to compute 2D capsules instead of their 3D approach. This allows for reduced computation and faster training compared to 3D capsules. Once the capsules compute the features, we get the class prediction from $ClassCaps$, which will have output for N classes. The pose information from the capsule layer is used to upsample using 3D transposed convolution to get final localization output of T frames with size $H \times W$. We also use skip connections and concatenate with the upsampled features to preserve features from the encoder head. The latent features used for clustering loss is taken from $Latent\_feats$ layer. The full architecture detail is shown in table 11

For the UCF-101-24 and J-HMDB-21 training, we use input clip of size $8 \times 224 \times 224 \times 3$ and get localization output for all 8 frames at size 224×224. We use variating skip rate of 1 and 2 to select the frames in each clip. We use a batch size of 8 for the training. The latent features from $Latent\_feats$ is used for computing the cluster loss.

7.2. Clustering details

We use the features from $Latent\_feats$ layer from the network as feature vector for clustering. Following [2, 8], we initialize the cluster with $K = 5$ centers initially using

%Frame Annot	Cost	video-mAP@ 0.1	0.2	0.3	0.5	frame-mAP@ 0.1	0.2	0.3	0.5
0.25	1000	94.91	87.98	78.34	45.00	86.59	82.03	74.92	50.11
0.50	2000	97.24	92.12	85.00	54.34	90.84	86.87	80.71	56.59
0.75	3000	97.65	93.63	88.25	57.68	91.61	88.70	83.12	59.49
1.00	4000	98.12	95.85	88.91	61.84	93.62	90.91	85.80	61.63
1.25	5000	98.25	95.95	89.97	65.55	93.42	90.94	86.44	65.61
1.50	6000	98.25	95.97	90.10	67.21	93.46	90.95	86.71	66.90
2.00	7000	98.28	95.6	90.92	68.64	93.20	90.99	87.15	68.54
2.50	8000	98.23	96.32	91.24	69.26	93.68	91.56	87.97	69.32
5.00	16000	98.33	96.35	91.25	72.28	94.28	92.28	88.38	72.12
90	362953	98.42	97.05	91.44	73.62	95.88	93.71	88.74	73.03
100	403282	99.28	97.86	91.54	75.29	98.79	96.69	89.13	74.07

Table 6. Evaluation of the proposed method on **UCF-101-24**. We increase the amount of samples and frames in each stage using the proposed approach and compare with fully-supervised approach.

%Frame Annot	Cost	video-mAP@ 0.1	0.2	0.3	0.5	frame-mAP@ 0.1	0.2	0.3	0.5
0.15	34	92.41	84.01	63.81	4.69	67.97	61.96	54.12	27.19
0.30	68	99.60	96.71	88.17	41.61	97.08	94.01	85.34	45.31
0.45	102	99.80	96.75	91.00	52.54	97.22	94.77	88.53	54.75
0.60	136	99.85	96.75	91.36	56.00	97.09	94.30	89.36	60.47
0.75	170	99.83	96.82	91.33	57.61	97.06	94.36	89.19	60.93
0.90	204	99.80	96.86	91.48	58.39	97.46	94.28	89.68	61.77
1.20	272	99.86	96.86	91.54	61.34	97.76	94.58	89.59	62.67
1.50	340	99.85	97.94	92.19	63.73	97.78	94.62	90.11	64.02
5.40	1226	99.87	98.35	95.24	71.50	98.26	94.98	92.33	72.85
90	20440	99.92	98.52	95.35	73.08	98.77	96.31	93.23	73.01
100	22712	99.98	99.01	96.40	75.81	99.05	97.84	93.70	74.93

Table 7. Evaluation of the proposed method on **J-HMDB-21**. We increase the amount of samples and frames in each stage using the proposed approach and compare with fully-supervised approach.

the latent features from the model for the current training set. During the training process, we use cluster loss to reduce the distance between the latent representation with the assigned cluster for each sample. For each active learning step, we use the learned cluster representation and assign cluster for each sample from the unlabeled set V_U. Based on the cluster assignment, we pick samples from different cluster for further labeling if the sample limit for that cluster has not crossed the threshold. This threshold stops oversampling from a single cluster.

7.3. Technical details

We run the training in a single Nvidia Quadro 5000 16GB GPU with a batch size of 8, with each sample with size $8 \times 224 \times 224 \times 3$ for $Frames \times Height \times Width \times Channels$. We use Adam optimizer [6] with learning rate of 0.0005 for 20 epochs in each active learning step. We also train the random selection variant for 20 epochs in each round. We use the weights from prior step to start the training for the next step. The model training time is reduced as only small fraction of annotated videos are used for training.

%Frame	video-mAP@				frame-mAP@			
Annot	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
Sample+Frame								
0.25	94.91	87.98	78.34	45.00	86.59	82.03	74.92	50.11
0.50	97.24	92.12	85.00	54.34	90.84	86.87	80.71	56.59
1.50	98.25	95.97	90.10	67.21	93.46	90.95	86.71	66.90
2.00	98.28	95.60	90.92	68.64	93.20	90.99	87.15	68.54
2.50	98.23	96.32	91.24	69.26	93.68	91.56	87.97	69.32
Only Frame								
0.25	94.91	87.98	78.34	45.00	86.59	82.03	74.92	50.11
0.50	97.24	92.12	85.00	54.34	90.84	86.87	80.71	56.59
1.50	97.28	92.38	85.65	59.33	91.28	86.99	81.03	61.20
2.00	97.62	93.33	86.31	62.42	91.57	87.23	82.95	62.83
2.50	97.63	93.45	89.30	66.03	92.49	88.55	85.14	66.94

Table 8. Comparing performance between sample + frame increment vs only frame increment for **UCF-101-24**. The *sample+frame* increment uses our method to increase V% samples in each active learning step and select F% frames from those samples for annotation. The *Only Frame* increment increases only F% annotation for an existing set of samples (without selecting V% more samples) in one of the active learning step.

Type	**Annot%**	**v-mAP@0.5**
Static	1%	69.2
Moving	1%	47.1
Static	5%	79.0
Moving	5%	59.9

Table 9. Evaluation of static vs moving actions on UCF-101-24. We report v-mAP @ 0.5 mIoU at 1% and 5% for each type of action.

References

[1] Hamed H Aghdam, Abel Gonzalez-Garcia, Joost van de Weijer, and Antonio M López. Active learning for deep detection neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3680, 2019. 4

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 6

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6, 11

[4] Kevin Duarte, Yogesh Rawat, and Mubarak Shah. Videocapsulenet: A simplified network for action detection. In *Advances in Neural Information Processing Systems*, pages 7610–7619, 2018. 6, 11

[5] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 4

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[7] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. 2019. 1

[8] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *international conference on machine learning*, pages 3861–3870. PMLR, 2017. 6

%Frame	video-mAP@				frame-mAP@			
Annot	0.1	0.2	0.3	0.5	0.1	0.2	0.3	0.5
STeW								
0.25	94.91	87.98	78.34	45.00	86.59	82.03	74.92	50.11
0.50	97.24	92.12	85.00	54.34	90.84	86.87	80.71	56.59
0.75	97.65	93.63	88.25	57.68	91.61	88.70	83.12	59.49
1.00	98.12	95.85	88.91	61.84	93.62	90.91	85.80	61.63
Interpolate								
0.25	92.03	95.90	77.75	43.82	84.26	79.11	72.39	48.23
0.50	97.09	92.56	84.16	51.64	92.22	88.11	80.33	53.66
0.75	97.84	94.02	85.70	53.85	92.97	90.05	81.71	54.47
1.00	98.13	94.15	86.38	58.22	93.90	90.31	82.86	58.86
Frame								
0.25	91.24	82.99	73.35	37.93	82.97	77.32	69.55	43.51
0.50	94.86	86.56	78.30	43.56	86.67	81.74	74.71	47.33
0.75	95.97	88.65	80.95	47.10	87.32	82.59	75.15	50.34
1.00	96.66	90.12	83.68	52.63	89.26	84.81	78.06	54.11

Table 10. Evaluating performance of different loss functions for the video action detection network for **UCF-101-24**. We compare model training for the proposed *STeW-Loss* with *Interpolation* loss and *Frame* level loss for various percent of frames annotated. For all loss variations we use the same *CLAUS* based active learning approach to increase annotations.

Figure 5. Demonstration of our proposed *STeW Loss* weights for interpolated pseudo-labels. Each sample shows the weight given to each pixel location based on the spatio-temporal consistency in pseudo-labels. The center frame is the frame with real ground-truth annotation and the red highlight shows the weight given with full weight given to center frame. We notice that the interpolated bounding box regions will have less consistency near the edges, so lower weight is given to those regions. Consistent foreground and background regions are given higher weight.

Layer	Name	Kernel Width (D x H x W)	Stride (D x H x W)	Output (D x H x W x C)
	Input			8 x 224 x 224 x 3
3D Conv	Conv3d_1a_7x7	7 x 7 x 7	2 x 2 x 2	4 x 112 x 112 x 64
3D Maxpool	MaxPool3d_2a_3x3	1 x 3 x 3	1 x 2 x 2	4 x 56 x 56 x 64
3D Conv	Conv3d_2b_1x1	1 x 1 x 1	1 x 1 x 1	4 x 56 x 56 x 64
3D Conv	Conv3d_2c_3x3	3 x 3 x 3	2 x 1 x 1	2 x 56 x 56 x 192
3D Maxpool	MaxPool3d_3a_3x3	1 x 3 x 3	1 x 2 x 2	2 x 28 x 28 x 192
3D Inception	Mixed_3b			2 x 28 x 28 x 256
3D Inception	Mixed_3c			2 x 28 x 28 x 480
3D Maxpool	MaxPool3d_4a_3x3	3 x 3 x 3	2 x 1 x 1	1 x 28 x 28 x 480
3D Inception	Mixed_4b			1 x 28 x 28 x 512
3D Inception	Mixed_4c			1 x 28 x 28 x 512
3D Inception	Mixed_4d			1 x 28 x 28 x 512
3D Inception	Mixed_4e			1 x 28 x 28 x 528
3D Inception	Mixed_4f			1 x 28 x 28 x 832
2D Conv Caps	Primary_caps			1 x 20 x 20 x 544
2D Conv Caps	Conv_caps			1 x 20 x 20 x 408
Latent Features	$Latent\_feats$			1 x 408
Class Caps	Class_caps			1 x 24
Poses Reshape	Poses			1 x 20 x 20 x 384
3D ConvTr	ConvTr_1	9 x 9 x 9	1 x 1 x 1	1 x 28 x 28 x 64
Concat	Concat_1			1 x 28 x 28 x 128
3D ConvTr	ConvTr_2	3 x 3 x 3	2 x 2 x 2	2 x 56 x 56 x 64
Concat	Concat_2			2 x 56 x 56 x 128
3D ConvTr	ConvTr_3	3 x 3 x 3	2 x 2 x 2	4 x 112 x 112 x 64
Concat	Concat_3			4 x 112 x 112 x 128
3D ConvTr	ConvTr_4	3 x 3 x 3	2 x 2 x 2	8 x 224 x 224 x 128
3D Conv	Conv3d_out	3 x 3 x 3	1 x 1 x 1	8 x 224 x 224 x 1

Table 11. Network architecture details. We use an I3D head for encoding the video information, followed by 2D capsules that will predict the class (24 for UCF-101-24) as class capsules. The pose information is then passed through a decoder with 3D transposed convolution for upsampling and concatenation with prior layers for skip connection. Each 3D Inception module follows the standard procedure from [3] and each Capsule layer follows [4].