t-EVA: Time-Efficient t-SNE Video Annotation

Soroosh Poorgholi(✉), Osman Semih Kayhan, and Jan C. van Gemert

Computer Vision Lab, Delft University of Technology, Delft, The Netherlands
s.poorgholi74@gmail.com, {O.S.Kayhan,J.C.vanGemert}@tudelft.nl

Abstract. Video understanding has received more attention in the past few years due to the availability of several large-scale video datasets. However, annotating large-scale video datasets are cost-intensive. In this work, we propose a time-efficient video annotation method using spatio-temporal feature similarity and t-SNE dimensionality reduction to speed up the annotation process massively. Placing the same actions from different videos near each other in the two-dimensional space based on feature similarity helps the annotator to group-label video clips. We evaluate our method on two subsets of the ActivityNet (v1.3) and a subset of the Sports-1M dataset. We show that t-EVA (https://github.com/spoorgholi74/t-EVA) can outperform other video annotation tools while maintaining test accuracy on video classification.

Keywords: Video annotation · t-SNE · Action recognition

1 Introduction

The availability of large-scale video datasets [17,19,20], has made video understanding in various tasks such as action recognition [25,37,41], object tracking [42,47,48] an attractive topic of research. Various supervised methods, [8,37,41], have improved video classification and temporal localization accuracy on large-scale video datasets such as ActivityNet (v1.3) [17]; however, labeling videos on such a large-scale dataset, requires a great deal of human effort. Therefore, other methods aim to train the networks for tasks such as video action recognition in a semi-supervised [1,46] manner without having the full labels. To decrease the dependency on the quality and amount of annotated data, [12,15] investigated pre-training features with internet videos with noisy labels in a weakly supervised manner. However, these methods cannot achieve higher accuracy on video classification tasks than supervised models on large-scale video datasets such as Kinetics [20]. Instead of using such techniques, we focus on reducing the annotation effort for adding more training data.

Fully-supervised models require much annotated data that is unavailable as videos are unlabeled by nature, and annotating them is labor-intensive. Large scale datasets [6,17,20] use strategies like *Amazon Mechanical Turk* (AMT) to annotate the videos. [20] uses majority voting between multiple AMT workers

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12664, pp. 153–169, 2021.
https://doi.org/10.1007/978-3-030-68799-1_12

to accept annotation of a single video. Using such methods is not efficient for video annotation on a large scale as it costs a lot in terms of time and money. MuViLab [2], an open-source software, enables the oracle to annotate multiple parts of a video simultaneously. However, these methods do not exploit the structure of the video data.

We introduce an annotation tool that helps the annotator group-label videos based on their latent space feature similarity in a 2-dimensional space. Transferring the high-dimensional features obtained from 3D ConvNet to two dimensions using t-SNE gives the annotator an easy view to group label the videos both, temporal labels and classification labels. The annotation speed depends on the quality of the extracted features and how well they are placed together in the t-SNE plot. If the classes are well-separated in the t-SNE plot, group labeling becomes faster for the oracle.

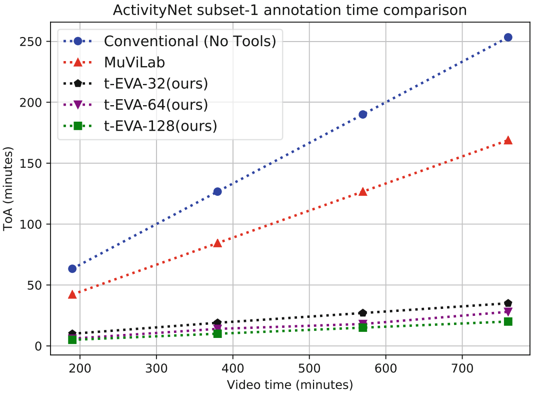


Fig. 1. Comparison of annotation time using different tools versus video time for the ActivityNet [17] subset-1. Our annotation method (t-EVA) outperforms the conventional (no specific tools) annotation and MuViLab [2] in annotation time. With a window size of 128 time-steps (128-TS), our method can annotate 769 min of video in 21 min. The MuViLab and conventional annotation numbers are extrapolated.

We evaluated our method on two subsets of ActivityNet (v1.3 datasets) [17] and a subset of Sports-1M dataset [19] with 15 random classes. *Conventional annotation* refers to humans watching the videos and annotating the temporal boundaries of the human actions in videos without any specific tool. *MuViLab* is a more advanced open-source tool that extracts short clips from each video and plays them simultaneously in a grid-like figure beside each other. Oracle can annotate the video by selecting multiple short clips at the same time and assigning the specific class. We show that t-EVA outperforms conventional annotation techniques (with no specific tools) and MuViLab [2] in time of annotation (ToA) by a large margin on the ActivityNet dataset while still being able to keep the test accuracy on video classification task within a close range of using the original ground truth annotations (Fig. 1).

2 Related Work

Video Understanding. In the past the focus was on the use of specific hand-designed features such as HOG3D [21] SIFT-3D [33], optical flow [34] and iDT [40]. Among these methods, iDT and Optical flow is being used in combination with CNNs in different architectures such as two-stream networks [36]. Later some attempts used 2D CNNs and extract features from video frames and combine them with different temporal integration functions [14,45]. The introduction of 3D convolution [35,37] in CNNs which extends the 2D CNNs in temporal dimension showed promising results in the task of action recognition in large-scale video datasets. 3D CNNs in different variations such as single stream and multiple-stream are among state of the art in the task of video understanding [4,10,13,18,28,32,38].

Dimensionality Reduction. Dimensionality reduction (DR) is an essential tool for high-dimensional data analysis. In linear DR methods such as PCA, the lower-dimension representation is a linear combination of the high-dimensional axes. Non-linear methods, on the other hand, are more useful to capture a more complex high-dimensional pattern [22]. In general, non-linear DR tries to maintain the local structure of the data in the transition from high-dimension to low-dimension and tends to ignore larger distances between the features [5]. t-Distributed Stochastic Neighbor Embedding (t-SNE) introduced by [39] is a non-linear DR technique which is used more for visualization. [24] shows that t-SNE is able to distinct well-separable clusters in low-dimensional space. Moreover, some works have been proposed for more effective use of t-SNE. [5] proposes a tool to support interactive exploration and visualization of high-dimensional data. An alternative to t-SNE is using UMAP [29] for dimensionality reduction. However, t-SNE is better studied, shows good results, and has the benefit of high-speed optimization [31]. Therefore, t-EVA uses t-SNE to reduce the dimensionality of the feature representations.

Data Annotation is essential for supervised models. Different tools have been proposed for making an easy annotation tool for videos and images. However, they usually do not exploit the structure of the data, which is especially useful in videos [2,3,7]. Some works [11,23,26,44] have been done to make the process of image annotation easier. [23] offers a real-time framework for annotating internet images, and [11] uses multi-instances learning to learn the classes and image attributes together; however, none of these methods use a deep representation of data. In more recent works [44] uses *Deep Multiple Instance Learning* to automatically annotate images and [26] uses semi-supervised t-SNE and feature space visualization in lower dimension to provide an interactive annotation environment for images. [9] proposed a general framework for annotating images and videos. However, to the best of our knowledge, our method is the first video annotation platform that can *exploit the structure* of video using latent space feature similarity to increase the annotation speed.

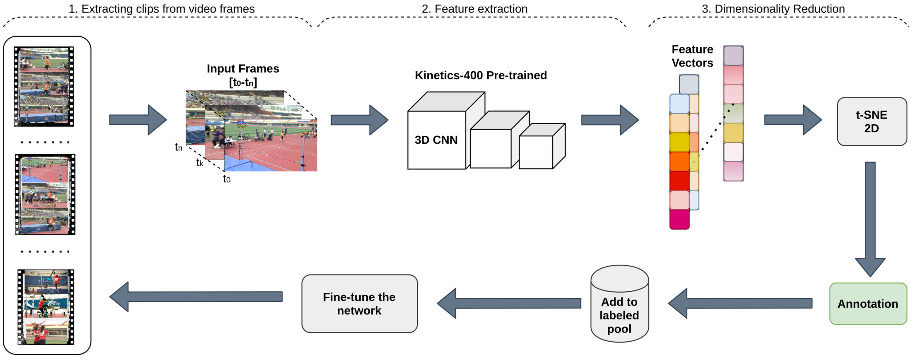


Fig. 2. t-EVA pipeline: 1) Video clips are extracted from n consecutive frames $[t_0$-$t_n]$ (time-steps). 2) Spatio-temporal features are extracted from the last layer of a 3D ConvNet before the classifier layer. 3) High dimensional features are projected to two dimensions using t-SNE and are plotted on a scatter plot. 4) Oracle annotates the clips represented in the scatter plot using a lasso tool. 5) The newly annotated data is added to the labeled pool. 6) The network is fine-tuned for a certain number of epochs. This cycle is repeated until all the videos are labeled, or the annotation budget runs out.

3 t-EVA for Efficient Video Annotation

We propose incremental labeling with t-SNE based on feature similarity (Fig. 2). First, several videos are randomly selected from the unlabeled pool, and 3D ConvNet features are extracted. The feature embeddings are transferred to a two-dimensional space using t-SNE. As it can be seen in Fig. 3, the oracle has two subplots for annotation: (i) A plot in which the oracle can use a lasso tool to group label videos and (ii) Other plot with the middle frame of each clip in which the oracle can move and zoom with the cursor on the plot and observe where to annotate. After annotating the first set of videos, the video clips are moved to the labeled pool, and the 3D network is fine-tuned for a certain number of epochs with the newly labeled videos. We continue this process until all the videos are labeled, or the annotation budget finishes.

We use 3D ConvNets to extract features from the videos and split each video v into k shorter clips $v_i = [clip_1, \ldots, clip_k]$ by sampling every n non-overlapping frames $clip_i = [frame_1, \ldots, frame_n]$. Sampling in multiple time-steps enables us to capture different lengths of actions in the dataset. Afterward, each clip c_i is fed into the 3D ConvNet, for feature extraction. The features are extracted from the last convolution layer after applying global average pooling. In t-SNE, the pair-wise distances between feature vectors are used to map features to 2D. In this paper, we use the Barnes-Hut optimized t-SNE version [27], which reduces the complexity of $O(NlogN)$ where N is the number of data-points.

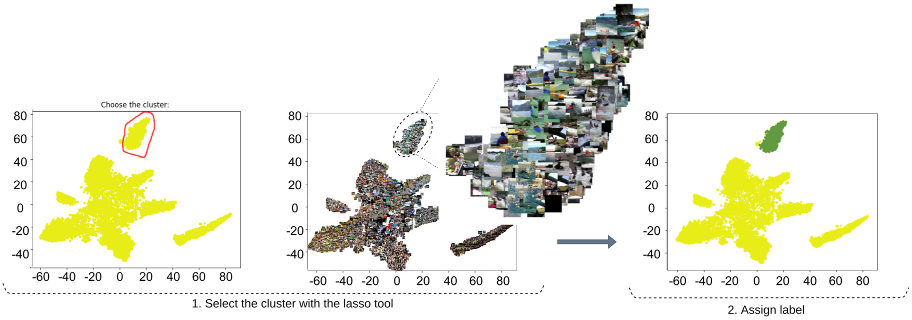


Fig. 3. A minimal representation of the annotation tool. 1) The oracle can see the scatter plot (left) and the corresponding frames from the videos (middle) in separate figures. 2) Based on the figures' inspection, the oracle can detect different clusters of an action class (kayaking) and use the lasso tool to select the cluster. 3) In the end, the oracle assigns a label and based on the assigned class name, the selected points in the scatter plot change color.

3.1 How to Annotate?

An overview of the annotation procedure can be seen in Fig. 3. First, the oracle sees the scatter plot with all points with the same color representing the unlabeled pool (Fig. 3 left) and the corresponding middle frame of each clip in the video (Fig. 3 middle). The oracle can move the cursor and zoom in the plot to inspect the frames with more details. Second, using the lasso tool, the oracle can draw a lasso around the scatter plots based on the visual similarity and inspection of the video frames. Third, oracle assigns the labels, and the network is fine-tuned for a certain number of epochs. The same process repeats until all the videos are annotated, or the annotation budget ends.

4 Experiments

In this section, we first explain the benchmark dataset and evaluation metrics. In addition, we empirically show how our t-EVA can speed up annotation for the ActivityNet dataset while keeping the video classification accuracy in a close range to the usage of the ground truth labels. We also compare our results with MuViLab [2] annotation tool. Furthermore, we qualitatively show how t-EVA can help to annotate the Sports1-M [19].

4.1 Datasets

ActivityNet (v1.3) is an untrimmed video dataset with a wide range of human activities [17]. It comprises of 203 classes with an average of 137 untrimmed videos per class in about 849 h of video. We use two subsets of the ActivityNet dataset. The first subset comprises 10 random classes, namely *preparing salad,* *kayaking, fixing bicycle, mixing drinks, bathing dog, getting a haircut, snatch, installing carpet, hopscotch, zumba* consisting of 607 videos with 407 training videos and 200 testing videos. The second subset adds another 5 handpicked classes, which are *playing water polo, high jump, discus throw, rock climbing, using parallel bars*, and they are visually close to some of the 10 random classes to make the classification task harder. The second subset comprises 950 videos with 639 videos in training and 311 videos in the test set.

Sports-1M is a large-scale public video dataset with 1.1 million YouTube videos of 487 fine-grained sports classes [19]. We choose a subset of 15 random classes of the Sports-1M dataset, namely *boxing, kyūdō, rings (gymnastics), yoga, judo, skiing, dachshund racing, snooker, drag racing, olympic weightlifting, motocross, team handball, hockey, paintball, beach soccer* with 702 videos in total. The dataset provides video level annotation for the entire untrimmed video; however, the temporal boundaries of the actions in the video are not identified. Approximately 5% of the videos contain more than one action label.

4.2 Evaluation Metrics

To evaluate our method on ActivityNet subsets, we report the *time of annotation* (ToA) as a metric to measure how fast the oracle can annotate a certain number of videos. The ToA score is an average of three times repeating each experiment by the oracle. ToA for conventional annotation and MuViLab on ActivityNet subset-1 is extrapolated since annotating 13 h of video using these methods is not feasible. We also report video classification accuracy in the form of mean average precision (mAP) for the ActivityNet subsets to measure the quality of annotation when the network is fine-tuned with our annotations versus with the ground truth annotations. mAP is used instead of a confusion matrix since some videos of ActivityNet contain more than one action [17].

For the Sports-1M [19] dataset, we perform a qualitative analysis of the t-SNE projections. To motivate our design choices beyond qualitative results, we introduce a realistic annotation emulation metric to estimate the quality of t-SNE projections on a global and local level. To report how well the t-SNE projection can separate the classes at a global level, we use a measure of cluster homogeneity, and completeness. Homogeneity measures if the points in a cluster only belong to one class and completeness measures if all points from one class are grouped in the same cluster. In an ideal t-SNE projection, all the points in each cluster belong to one class (homogeneity = 1.0), and all the points from a class are in the same cluster (completeness = 1.0), which makes the annotation process much faster. For clustering, K-Means clustering with K being the number of classes is used. We use the K-Means clustering algorithm because it is fast and has less hyperparameters to choose.

Since ToA can be a subjective metric, to evaluate the generalization of t-EVA and to emulate the oracle's annotation speed better, we also use a measure of local homogeneity using K-nearest neighbors (KNN) with $K = 4$ as in [26].

KNN can be used to estimate the local homogeneity between the features in lower dimensions. Higher KNN accuracy results in higher local homogeneity and better grouping; meaning, the oracle can annotate the videos faster.

4.3 Implementation Details

Feature Extraction. We use the 3D ResNet-34 architecture [16], pre-trained on Kinetics-400, as a feature extractor for all the experiments owing to their good performance and usage of RGB frames only. As in [16], each frame is resized spatially to 112×112 pixels from the original resolution. Each video is transferred to clips by sampling every 32 consecutive frames. The feature extractor in every forward pass takes a clip in the form of a 5D tensor as an input. Each dimension of the input tensor represents the batch size, input color channels, number of frames, spatial height, and width, respectively. Namely, an input tensor for a clip sampled at 32 frames can be shown as (1, 3, 32, 112, 112). The features are extracted after the final 3D average pooling with an $8 \times 4 \times 4$ kernel before the classifier layer. The dimensions of the feature vectors are $k \times 512$ with k being the total number of clips and later reduced to $k \times 2$ using t-SNE.

t-SNE. For dimensionality reduction, a Barnes-Hut implementation of t-SNE with two components are used from the scikit-learn library [30]. The perplexity is set to 30, and the early exaggeration parameter is 12, with a learning rate of 200. The cost function is optimized for 2500 iterations.

Training. After annotating each set of videos, the network is fine-tuned for a certain number of epochs. For training, the same 3D ResNet-34 [16] architecture is used. The sample duration is chosen as 32 frames for each clip, and the input batch size is 32. Stochastic gradient descends (SGD) is used as the optimizer with a learning rate of 0.1, weight decay of $1e{-}3$, and momentum of 0.9.

4.4 Results on ActivityNet

ActivityNet Subset-1. First, we put all the 407 videos in the unlabeled pool. Then, we divide the videos randomly into four different sets of unlabeled videos. The clips are generated with 32 consecutive frames, and the features are extracted using the 3D Resnet-34. After annotating each set of unlabeled videos, the network is fine-tuned for 20 epochs with the labeled videos. To note that, previously labeled videos are also used in the later epochs. The process continues until the network reaches 100 epochs. Between epoch 60 and 100, the network is fine-tuned using all 407 videos. Meanwhile, we refine the labels of the videos.

The videos are annotated incrementally, each time one set is labeled. Table 1 shows that the annotation time drops after every iteration of annotation and fine-tuning. Before fine-tuning the network, the labeling of the first set takes 600 s. ToA reduces 150 s at epoch 60 when the network is fine-tuned with previously labeled videos. Because of the incremental labeling and fine-tuning, the network learns to extract better features from the videos, which can be better grouped in the t-SNE plot. It is also expected that the oracle spends more time annotating the first few unlabeled set as the network is not yet fine-tuned. The quality of annotation at the early stage significantly impacts the next iterations of extracted features.

Table 1. Oracle's time of annotation (ToA) is shown on subset 1 of the ActivityNet (v1.3) dataset with 10 classes containing 407 videos (∼13 h). At every 20 iterations from 0 to 60, 102 new videos are annotated, and the network is fine-tuned for 20 epochs. From epoch 60 to 100, no new video is added. The previous video labels are refined by the oracle as the network can extract better features. The network is fine-tuned on the existing labeled videos until epoch 100. It can be seen with incremental annotation and fine-tuning the annotation time in the later epochs drops.

Epoch	0	20	40	60	80	100
ToA (s)	600	552	516	450	240	180

Annotation Speed. To evaluate the annotation speed, we choose three methods: conventional, MuViLab [2], and t-EVA.

One way to increase the annotation speed of t-EVA is by putting more videos on the screen for the oracle to annotate. However, it does not make the labelling process easier. Since ActivityNet videos on average have 30 frames per second (FPS), every 32 time-steps that we sample represent almost 1 s ($\sim\frac{32}{30}$) of video. Putting all of the 407 videos (13 h) overflows the screen with the frames and makes the annotation harder for the oracle. One way to prevent overflowing the figures with thousands of frames is to increase the time-steps for sampling frames from each clip to the point that the network can still preserve the clips' temporal coherency. This way, we can show all of the videos on the 2D plot with fewer points. Consequently, we design three different t-EVA in terms of the number of time steps as t-EVA-32, 64, and 128.

Table 2. Comparison of time gain when annotating with different methods on a subset-1 of ActivityNet containing 769 min of video. Our method (t-EVA) with 128 time-steps outperforms conventional, and MuViLab [2] methods with labeling 769 min of video in 21 min. Using more consecutive frames increases annotation speed.

	Conventional	MuViLab	t-EVA-32	t-EVA-64	t-EVA-128
Time Gain	3×	4.5×	18×	24×	36×

First, we choose ActivityNet subset-1 with a total duration of 769 min. We annotated 30 min of videos using MuViLab and Conventional methods and extrapolated the result to match the total duration of ActivityNet subset-1. Additionally, the entire subset-1 is annotated using different variants of t-EVA, and we compare the annotation speed of all these methods (Table 2). The results

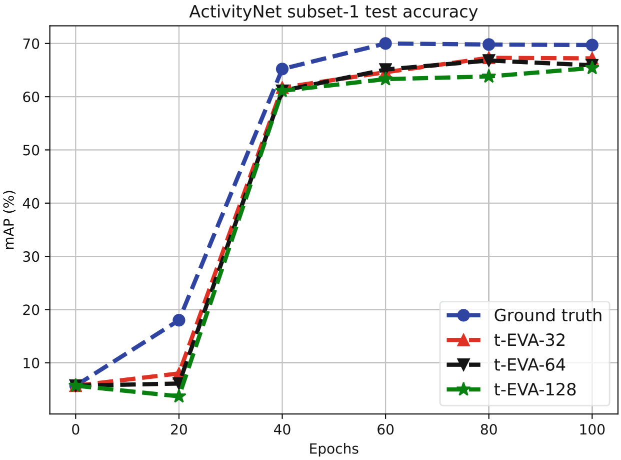


Fig. 4. Comparison of video classification performance in the form of mAP (%) between fine-tuning the 3D ConvNet on ground truth label versus fine-tuning with our annotation acquired using different time-steps (TS). Fine-tuning the 3D ConvNet on the annotation generated by our method can achieve comparable video classification accuracy to the ground truth.

show that labeling 769 min of video takes approximately 21 min with the t-EVA-32 method. t-EVA-32 outperforms both conventional and MuViLab methods on ActivityNet subset-1 in annotation speed by a large margin by respectively 4 to 6 times faster. With t-EVA-64 and 128, time gain can reach respectively 24 and 36 times more. Conventional annotation and MuViLab do not take advantage of the temporal dimension of videos for annotation. Nevertheless, our method exploits the spatio-temporal features and places similar actions near each other in the t-SNE plot for the oracle to annotate the actions.

We also evaluate the performance of the network on the test set of ActivityNet subset-1. In Fig. 4, we compare the classification performance of the networks: (i) fine-tuned with original ground truth labels and (ii) fine-tuned by using newly annotated videos by 32, 64, and 128 time-steps. Annotating the videos with t-EVA method can achieve a classification performance of 67.2% with 32-TS, 65.9% with 64-TS, and 65.4% with 128-TS, which is comparable to the training with ground truth labels (blue) by 69.7% mAP.

Table 3 shows the speed-accuracy trade off between t-EVA and ground-truth annotation. When the original ground truth labels are used for fine-tuning the network, we obtain 69.7% of mAP. 407 videos can be labeled in 42 min with t-EVA-32 by losing only 2.5% of performance in comparison to using ground truth labels. When the time-steps are increased as 64 and 128, the annotation speed decreases respectively to 31 and 21 min, yet the classification performance also reduces by 3.8% and 4.3%. Using 128 time-steps (t-EVA-128) reduces test accuracy while increasing the annotation speed. The decrease in accuracy compared to the 32-TS version is expected since the annotation is more prone to noise when the time-step is increased to 128 frames. With 128-TS for each clip, every point in the scatter plot represents 4 s of the video while it represents 1 s in the 32-TS version. Namely, labeling points wrongly in the 128 version (t-EVA-128) brings more significant consequences in the fine-tuning process. However, Table 3 indicates that using 128-TS (t-EVA-128) compared to the 32-TS (t-EVA-32) increases the annotation speed twice while the mAP score decreases less than 2%.

Table 3. Comparison of video classification performance (mAP) and ToA (time of annotation) on ActivityNet subset-1. This subset contains 407 videos in about 13 h of video. Our method in 32 time-steps (t-EVA-32) and 128 time-steps (t-EVA-128) achieves comparable test accuracy to the ground truth accuracy and requires a much shorter time to annotate. There is a trade-off between annotation speed and performance.

Method	GT	t-EVA-32	t-EVA-64	t-EVA-128
mAP	69.7%	67.2%	65.9%	65.4%
ToA (min)	–	42	31	21

4.5 Generalization

To further demonstrate the generalization of our method, we conduct the same annotation experiment on a more challenging subset of ActivityNet (v1.3) with 15 classes and a subset of Sports-1M [19] with 15 random classes.

ActivityNet (v1.3) Subset-2. Subset 2 of ActivityNet (v1.3) contains 637 training videos and 311 test videos. The first iteration of features is extracted from the 637 training videos and is annotated in 15 min by the oracle using t-EVA. After 20 epochs of fine-tuning, the new features are extracted, and the labels are fine-tuned again by the oracle. After this stage, the network is fine-tuned for 80 epochs. After fine-tuning for 100 epochs, our method reaches a test accuracy of 66.4%, while the training with ground-truth labels achieves an accuracy of 68.3% on the video classification task.

The 4-NN accuracy of the final features is 92.4%, which shows the quality of the extracted features is sufficient for the oracle to annotate. t-EVA can also perform well on the ActivityNet subset-2. The fact validates that our method can also generalize on a more challenging subset of ActivityNet.

Sports-1M. We further validate our method on a subset of Sports-1M [19] dataset with 15 random classes. We randomly sample 200 videos (∼860 min) from the total 702 videos available in the 15 classes. The features are extracted from 200 videos, and ground truth labels of the two-dimensional features can be seen in Fig. 5. Using 4-NN, we obtain an accuracy of 92.3%, which shows the features can be annotated based on similarity. Using our method, we were able to annotate 860 min of video in 28 min, giving us a time gain of 30.7. t-EVA indicates an extensive time gain on the Sports-1M dataset.

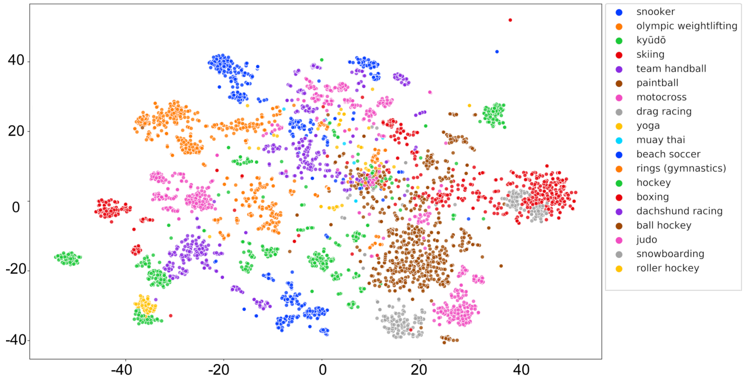


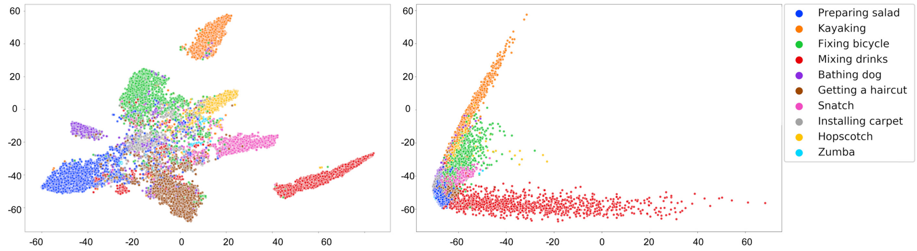
Fig. 5. t-SNE projection of extracted features from 200 videos from the Sports-1M [19] dataset with ground truth labels as colors. 200 videos are from 15 random classes; however, some videos contain more than one activity class. The 4-NN accuracy, which emulates the quality of the projection through measuring local homogeneity, is 92.3%, indicating such a figure is annotate-able by the oracle.

5 Ablation Study

In this section, we conduct an ablation study to motivate our design choices in the following aspects: (i) dimensionality reduction method, (ii) t-SNE parameter selection, and (iii) 2D versus 3D backbone for feature extraction.

5.1 Dimensionality Reduction

We investigate using PCA as a linear dimensionality method and t-SNE as a non-linear dimensionality method for visualizing the high-dimensional features in two dimensions. We use the extracted feature from the ActivityNet subset-1 with 407 videos. Figure 6-b shows qualitatively that PCA is not able to group similar features and separate unalike features from the videos in the transition to a lower dimension, making the annotation more difficult. However, Fig. 6-a, shows that t-SNE projection can maintain the local structure of each class while separating the features from different classes. To report the quality of projection in quantitative measures, we use KNN with $K = 4$. The 4-NN classification accuracy in Fig. 6 for the t-SNE projection is 80.6%, and for the PCA projection is 58.2%. Therefore, PCA, a linear dimensionality method, cannot reduce the feature dimension while placing similar classes near each other.



a) t-SNE projection

b) PCA projection

Fig. 6. Visual comparison of the projection quality of high-dimensional features to two dimensions using t-SNE (a) and PCA (b). PCA is unable to maintain the structure of the high-dimensional data in two dimensions.

5.2 t-SNE Parameters

We investigate using different perplexity parameters for the t-SNE projection. [39] recommend using perplexity parameter between [5–50], however larger and denser datasets requires relatively higher perplexity. With low perplexity, the local structure of data in each video dominates the action grouping from multiple video [43], but our goal is to group multiple actions from different videos. To emulate the t-SNE projection quality for the annotation, we report homogeneity and completeness scores with different perplexities in Table 4. Perplexity 30 shows the highest homogeneity and completeness scores, meaning that t-SNE projection with perplexity 30 can separate the classes better than projecting with the other perplexity parameters. Therefore, using t-SNE with perplexity 30 makes the group labeling process easier for the oracle.

5.3 2D-3D Comparison

We investigate replacing the 3D ConvNet with a 2D CNN to compare the quality of the feature embedding. For 3D ConvNet, 3D ResNet-34 pre-trained on Kinetics [20] and for the 2D CNN ResNet-50 pre-trained on Kinetics [20] are used. We chose Resnet-50 instead of Resnet-34 for the 2D CNN because the Kinetics pre-trained weights were only available for ResNet-50. To experiment, we sample every 32 consecutive frames (time-steps) as a clip in the 3D ConvNet, and for the 2D CNN, we choose one frame for every 32 frames to represent that specific window. The experiment is done on the subset-1 of the Activity-Net dataset with 10 classes. It can be seen in Fig. 7 that we start the experiments with 32 time-steps. With 32 time-steps, we can see the 2D CNN can capture the same action in different videos but can not place them together as well as the 3D ConvNet. Therefore, the colors representing the classes are better gathered

Table 4. Comparison of homogeneity and completeness scores as a measure to emulate the quality of t-SNE projection on a global-level. Higher homogeneity means all the points in a cluster belong to the same class. Higher completeness means all the points belonging to a class are in the same cluster. t-SNE perplexity parameter as 30 gives the highest homogeneity and completeness score.

	px-5	px-15	px-30	px-50	px-100	px-120
Homogeneity	44.7%	58.7%	62.5%	61.3%	61.7%	61.5%
Completeness	42.5%	56.1%	60%	58.5%	59%	58.8%

nearby in the 3D ConvNet, making the annotation process faster than the 2D CNN projection. Moreover, by increasing the time-steps for frame sampling, the 2D CNN, even with deeper architecture, starts losing the temporal coherency between the data-points because 2D CNN only focuses on the spatial information between the frames. Focusing only on spatial information can still work in lower time-steps (32-TS) because the frames from the same action contain similar spatial information. However, using spatial information alone becomes problematic in higher time-steps as increasing the time-steps reduces the spatial similarity between the frames.

To evaluate our findings quantitatively, we use K-NN accuracy as a quantitative emulation for the quality of features for annotation. Table 5 shows that increasing the number of frames in the clips degrades the 4-NN accuracy of 2D CNN dramatically from 93% to 75%. However, 3D CNN only loses around 5% from 32 time steps to 128. The local homogeneity decreases more drastically in 2D CNNs compared to 3D CNNs, which makes annotation more difficult for the oracle. In other words, the 2D CNN alone can not maintain the temporal structure of the data in higher time-steps. Thus, in the t-EVA method, 3D features are extracted to use for group labeling.

Table 5. Comparison of 4-NN accuracy of extracted features from a 2D CNN (ResNet-50) and a 3D ConvNet (3D ResNet-34) on subset-1 of ActivityNet [17]. Increasing time-steps cause the 2D CNN to lose the spatial similarity between the frames and fail to group them in the t-SNE plot, while the 3D ConvNet can still group similar actions even in higher time-steps.

	32-TS	64-TS	128-TS
2D CNN	93.1%	89.3%	74.6%
3D CNN	100%	97.6%	95.2%

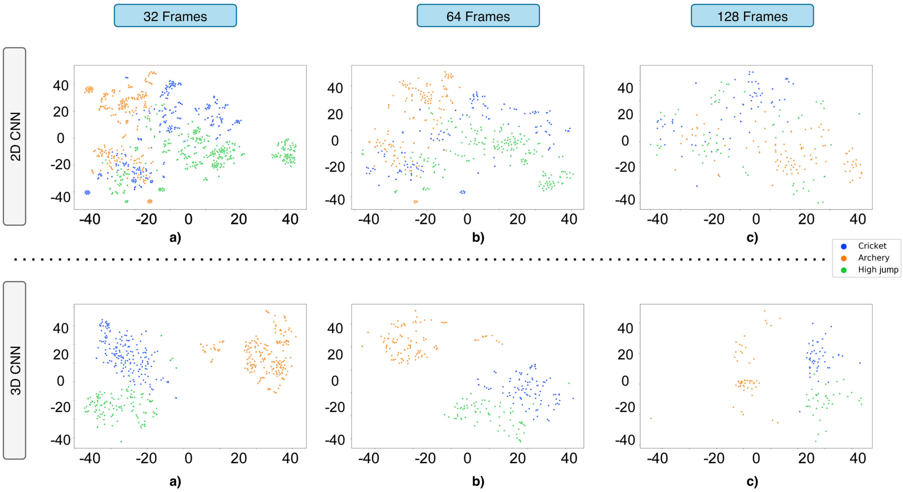


Fig. 7. Comparison of t-SNE projection of extracted features from a 2D CNN versus a 3D ConvNet for videos from 3 action classes of ActivityNet dataset [17]. Increasing the time-steps for sampling clips from the videos causes the 2D CNN to lose the clips' spatial information. However, the features from the 3D ConvNet can maintain the coherency between the clips.

6 Conclusion

This paper introduced a smart annotation tool, t-EVA, for helping the oracle to group label videos based on their latent space feature similarity in two-dimensional space. Our experiments on subsets of large-scale datasets shows that t-EVA can be useful in annotating large-scale video datasets, especially if the annotation budget and time are limited. Our method can outperform the conventional annotation method, and MuViLab [2] time-wise in the order of magnitude with a minor drop in the video classification accuracy. Besides, t-EVA is a modular tool, and its components can be easily replaced by other methods. To illustrate, 3D ResNet can be changed to another feature extractor.

t-EVA method has a trade-off between annotation speed and network performance. Increasing time steps can reduce the annotation time; however, the network's accuracy may also decrease.

t-EVA can be sensitive to the initial state of the feature extractor. If the feature extractor can not separate classes well, it can take a longer time to annotate the videos initially. After fine-tuning the network with new labels for a few epochs, the labeling time can reduce again. Besides, putting more video frames in the t-SNE plot can overflow the screen and make the annotation process harder for the oracle.

References

1. Ahsan, U., Sun, C., Essa, I.A.: DiscrimNet: semi-supervised action recognition from videos using generative adversarial networks. CoRR abs/1801.07230 (2018). http://arxiv.org/abs/1801.07230
2. Alessandro Masullo, L.D.: Muvilab (2019). https://github.com/ale152/muvilab
3. Gupta, A.K.: ImgLab (2017). https://github.com/NaturalIntelligence/imglab
4. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. CoRR abs/1705.07750 (2017). http://arxiv.org/abs/1705.07750
5. Chatzimparmpas, A., Martins, R.M., Kerren, A.: t-viSNE: interactive assessment and interpretation of t-SNE projections. IEEE Trans. Vis. Comput. Graph. **26**(8), 2696–2714 (2020). https://doi.org/10.1109/tvcg.2020.2986996
6. Damen, D., et al.: Scaling egocentric vision: the EPIC-KITCHENS dataset. CoRR abs/1804.02748 (2018). http://arxiv.org/abs/1804.02748
7. darrenl, o.c.: labelimg (2017). https://github.com/tzutalin/labelImg
8. Diba, A., et al.: Temporal 3d ConvNets: new architecture and transfer learning for video classification. CoRR abs/1711.08200 (2017). http://arxiv.org/abs/1711.08200
9. Dutta, A., Zisserman, A.: The via annotation software for images, audio and video. In: Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, pp. 2276–2279. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3343031.3350535
10. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast networks for video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
11. Wang, G., Forsyth, D.: Joint learning of visual attributes, object classes and visual saliency. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 537–544 (2009)
12. Ghadiyaram, D., Feiszli, M., Tran, D., Yan, X., Wang, H., Mahajan, D.: Large-scale weakly-supervised pre-training for video action recognition. CoRR abs/1905.00561 (2019). http://arxiv.org/abs/1905.00561
13. Girdhar, R., Carreira, J., Doersch, C., Zisserman, A.: Video action transformer network. CoRR abs/1812.02707 (2018). http://arxiv.org/abs/1812.02707
14. Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: ActionVLAD: learning spatio-temporal aggregation for action classification. In: Proceedings of (CVPR) Computer Vision and Pattern Recognition, pp. 3165–3174 (July 2017)
15. Girdhar, R., Tran, D., Torresani, L., Ramanan, D.: Distinit: Learning video representations without a single labeled video. CoRR abs/1901.09244 (2019). http://arxiv.org/abs/1901.09244
16. Hara, K., Kataoka, H., Satoh, Y.: Learning spatio-temporal features with 3d residual networks for action recognition. CoRR abs/1708.07632 (2017). http://arxiv.org/abs/1708.07632
17. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: ActivityNet: a large-scale video benchmark for human activity understanding. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 961–970 (2015)
18. Hussein, N., Gavves, E., Smeulders, A.W.M.: Timeception for complex action recognition. CoRR abs/1812.01289 (2018). http://arxiv.org/abs/1812.01289
19. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)

20. Kay, W., et al.: The kinetics human action video dataset. CoRR abs/1705.06950 (2017). http://arxiv.org/abs/1705.06950
21. Kläser, A., Marszalek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC (2008)
22. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Springer, New York (2007). https://doi.org/10.1007/978-0-387-39351-3
23. Li, J., Wang, J.Z.: Real-time computerized annotation of pictures. IEEE Trans. Pattern Anal. Mach. Intell. **30**(6), 985–1002 (2008)
24. Linderman, G.C., Steinerberger, S.: Clustering with t-SNE, provably. CoRR abs/1706.02582 (2017). http://arxiv.org/abs/1706.02582
25. Liu, K., Liu, W., Gan, C., Tan, M., Ma, H.: T-C3D: temporal convolutional 3d network for real-time action recognition. In: AAAI (2018)
26. Luus, F.P.S., Khan, N., Akhalwaya, I.: Active learning with TensorBoard Projector. CoRR abs/1901.00675 (2019). http://arxiv.org/abs/1901.00675
27. van der Maaten, L.: Barnes-Hut-SNE (2013)
28. Martinez, B., Modolo, D., Xiong, Y., Tighe, J.: Action recognition with spatial-temporal discriminative filter banks (2019)
29. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction (2018)
30. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
31. Pezzotti, N., et al.: GPGPU linear complexity t-SNE optimization. IEEE Trans. Visual. Comput. Graph. (Proc. VAST 2019) **26**(1), 1172–1181 (2020). http://graphics.tudelft.nl/Publications-new/2020/PTMHLLEV20
32. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. CoRR abs/1711.10305 (2017). http://arxiv.org/abs/1711.10305
33. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM International Conference on Multimedia, MM 2007, pp. 357–360. Association for Computing Machinery, New York (2007). https://doi.org/10.1145/1291233.1291311
34. Sevilla-Lara, L., Liao, Y., Güney, F., Jampani, V., Geiger, A., Black, M.J.: On the integration of optical flow and action recognition. CoRR abs/1712.08416 (2017). http://arxiv.org/abs/1712.08416
35. Shou, Z., Wang, D., Chang, S.: Action temporal localization in untrimmed videos via multi-stage CNNs. CoRR abs/1601.02129 (2016). http://arxiv.org/abs/1601.02129
36. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. CoRR abs/1406.2199 (2014). http://arxiv.org/abs/1406.2199
37. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: C3D: generic features for video analysis. CoRR abs/1412.0767 (2014). http://arxiv.org/abs/1412.0767
38. Tran, D., Wang, H., Torresani, L., Feiszli, M.: Video classification with channel-separated convolutional networks. CoRR abs/1904.02811 (2019). http://arxiv.org/abs/1904.02811
39. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-SNE. J. Mach. Learn. Res. **9**, 2579–2605 (2008)
40. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: 2013 IEEE International Conference on Computer Vision, pp. 3551–3558 (2013)
41. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. CoRR abs/1608.00859 (2016). http://arxiv.org/abs/1608.00859

42. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.S.: Fast online object tracking and segmentation: a unifying approach. CoRR abs/1812.05050 (2018). http://arxiv.org/abs/1812.05050
43. Wattenberg, M., Viégas, F., Johnson, I.: How to use t-SNE effectively (2016)
44. Wu, J., Yu, Y., Huang, C., Yu, K.: Deep multiple instance learning for image classification and auto-annotation (June 2015)
45. Xu, Z., Yang, Y., Hauptmann, A.G.: A discriminative CNN video representation for event detection. CoRR abs/1411.4006 (2014). http://arxiv.org/abs/1411.4006
46. Zeng, M., Yu, T., Wang, X., Nguyen, L.T., Mengshoel, O.J., Lane, I.: Semi-supervised convolutional neural networks for human activity recognition. CoRR abs/1801.07827 (2018). http://arxiv.org/abs/1801.07827
47. Zhang, Y., Wang, C., Wang, X., Zeng, W., Liu, W.: A simple baseline for multi-object tracking (2020)
48. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware Siamese networks for visual object tracking. CoRR abs/1808.06048 (2018). http://arxiv.org/abs/1808.06048