Incremental Activity Modeling and Recognition in Streaming Videos

Mahmudul Hasan and Amit K. Roy-Chowdhury
University of California, Riverside
mhasa004@ucr.edu, amitrc@ee.ucr.edu

Abstract

Most of the state-of-the-art approaches to human activity recognition in video need an intensive training stage and assume that all of the training examples are labeled and available beforehand. But these assumptions are unrealistic for many applications where we have to deal with streaming videos. In these videos, as new activities are seen, they can be leveraged upon to improve the current activity recognition models. In this work, we develop an incremental activity learning framework that is able to continuously update the activity models and learn new ones as more videos are seen. Our proposed approach leverages upon state-of-the-art machine learning tools, most notably active learning systems. It does not require tedious manual labeling of every incoming example of each activity class. We perform rigorous experiments on challenging human activity datasets, which demonstrate that the incremental activity modeling framework can achieve performance very close to the cases when all examples are available a priori.

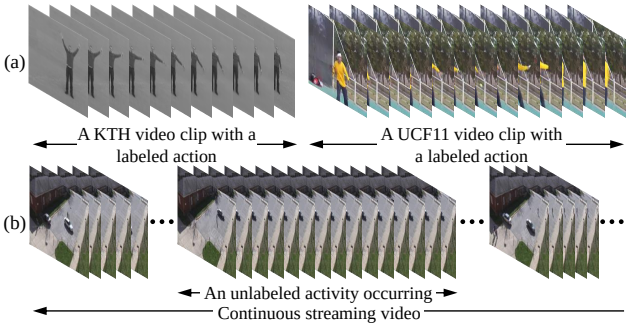

Figure 1: The top row (a) shows two video clips of KTH [1] and UCF11 [2] human activity dataset and the bottom row (b) shows a video stream of VIRAT ground human activity database [3]. Most of the state-of-the-art approaches perform well on activity datasets like KTH and UCF11, where they assume that one video clip contains only one labeled action, their temporal extent is known, and all of the training examples are available beforehand. However, in continuous video stream like VIRAT, new activities may arrive after the training stage, which are usually unlabeled and their spatio-temporal extent is unknown.

1. Introduction

Human activity recognition is a challenging and widely studied problem in computer vision. It has many practical applications such as video surveillance, video annotation, video indexing, active gaming, human computer interaction, assisted living for elderly, etc. Even though enormous amount of research has been conducted in this area, it still remains a hard problem due to large intra-class variance among the activities, large variability in spatio-temporal scale, variability of human pose, periodicity of human action, etc. Low quality video, clutter, occlusion, etc. also add more difficulties to the problem.

With few exceptions, most of the state-of-the-art approaches [4] to human activity recognition in video are based on one or more of the following four assumptions: (a) It requires an intensive training phase, where every training example is assumed to be available; (b) Every training example is assumed to be labeled; (c) At least one example of every activity class is assumed to be seen beforehand, i.e., no new activity type will arrive after training; (d) A video clip contains only one activity, where the exact spatio-temporal extent of the activity is known. However, these assumptions are too strong and not realistic in many real world scenarios such as streaming and surveillance videos. In these cases, new unlabeled activities are coming continuously and the spatio-temporal extent of these activities are usually unknown in advance, as explained in Figure 1.

Motivated by the above, the **main goal** of this work is twofold: to classify new unknown activities in streaming videos, and also leverage upon them to continuously improve the existing activity recognition models. In order to achieve this goal, we develop an incremental activity learning framework that will use new activities identified in the incoming video to *incrementally improve* the existing models by leveraging novel machine learning techniques, most

This work was supported in part by ONR grant N00014-12-1-1026 and NSF grant IIS-1316934.

Mahmudul Hasan is with Dept. of Computer Science and Amit K. Roy-Chowdhury is with Dept. of Electrical Engineering at UCR.

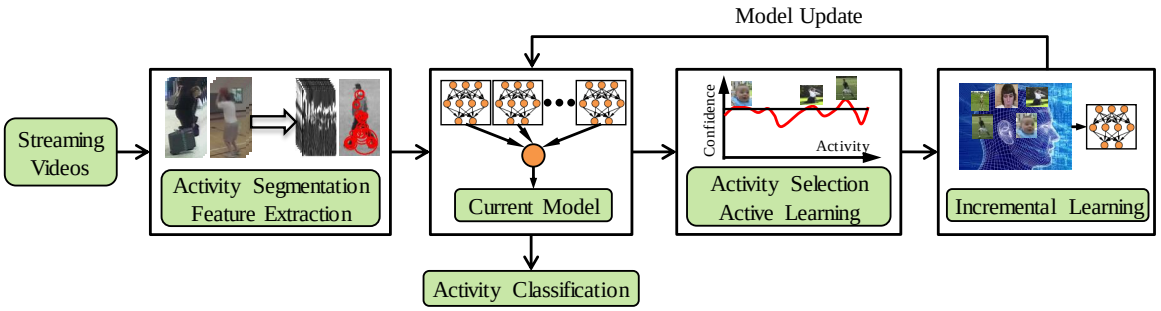

Figure 2: This figure shows our proposed incremental activity modeling framework, which is comprised of following stages: activity segmentation, feature extraction, model learning, activity classification, training activity selection by active learning, and model updating by incremental learning with the help of the active learning system.

notably active learning.

The detailed framework of our proposed incremental activity recognition algorithm is shown in Figure 2. Since, we do not have any prior information about the spatio-temporal extent of the activities in the continuous video, our approach begins with video segmentation and localization of the activities using a motion segmentation algorithm. Each of the atomic motion segments are considered as the activity segments from which we collect spatio-temporal features such as STIP [5], higher level features such as Action Bank (AB) [6], and global features such as Gist3D [7]. These features are widely used in action recognition and achieve satisfactory performance in state-of-the-art challenging datasets. Then, we learn a prior model using few labeled training activities in hand. In this work, we propose to use an ensemble of linear Support Vector Machine (SVM) classifiers as the prior model. Note that *we do not assume that the prior model is exhaustive* in terms of covering all activity classes or in modeling the variations within the class. It is used only as a starting point for the incremental learning framework.

We start incremental learning with the above mentioned prior model and update it during each run of incremental training. When a newly segmented activity arrives, we apply the current model to get a tentative label with a confidence score. However, it is not practical and rational to use all of the newly segmented activities as the training examples for the next run of incremental training. This is because it is costly to get a label for all of them from a human annotator, and not all of them posses distinguishing properties for effective update of the current model. We only select a subset of them and rectify the tentative labels by our proposed active learning system. In order to learn the activity model incrementally, we employ an ensemble of linear SVMs. When we have sufficient new training examples labeled by the active learning system, we train a new set of SVM classifiers and consequently, update the current model by adding these new SVM classifiers to the ensemble with appropriate weights.

Thus, the **main contribution** of this work is to develop a framework to incrementally learn activity models from streaming videos, which is achieved through an active learning framework. This includes updating already known models with new examples, as well as learning new classes. This will reduce tedious manual labeling that is needed for most state-of-the-art systems. We assume that the total number of classes is known.

2. Related Works

We would like to refer to the article [4] for a comprehensive review on the sate-of-the-art approaches to human activity recognition. Based on the level of abstraction used to represent an activity, state-of-the-art approaches can be classified into three general categories such as low-level [5], mid-level [8], and high-level [6] feature based methods. In some recent works, graphical models [9], AND-OR grammar [10], and contextual information surrounding the activity of interest [11] are exploited for modeling more complex activities. However, as discussed in Section 1, most of these state-of-the-art approaches suffer from the inability to model activities in continuous streaming video and unable to take advantages of unseen incoming activities.

Incrementally learning from streaming data is a well studied problem in machine learning and a lot of approaches have been proposed in the literature. Among these approaches, ensemble of classifiers [12,13] based methods are most commonly used, where new weak classifiers are trained as new data is available and added to the ensemble. Their outputs are combined using an appropriate combination rule, which is set according to the system's goal.

Active learning has been successfully used in speech recognition, information retrieval, and document classification [14]. Some recent works used two stage active learning framework in several computer vision applications such as image segmentation [15], image and object classification [16], unusual event detection [17], action recognition [18], etc. However, unlike most of these methods,

our framework does not require the storage of already used training examples and takes the advantage of highly confident decision provided by the current classification model, which in turns reduces the amount of manual labeling.

A few methods have considered incremental activity modeling. A feature tree based incremental action recognition method was proposed in [19], where the feature-tree grows when additional training examples are available. It requires the storage of all training examples in the form of feature tree, which is not feasible for continuous streaming videos because the number of activities could be very large over time. Human track-based incremental activity learning framework was proposed in [8]. It requires annotation of the human body in the initial frame of an action clip, which restricts the variety of application domains possible.

3. Incremental Activity Modeling Methodology

We now provide a detailed overview of our proposed incremental activity modeling framework.

3.1. Activity Segmentation and Feature Extraction

We use an adaptive background subtraction algorithm [20] to locate motion regions in the continuous video. Inside these motion regions, moving persons are detected by [21]. These detections are used to initialize the tracking method developed in [22] to obtain local trajectories of the moving persons. Spatio-temporal interest points (STIP) [5] are collected only for these motion regions. Each motion region is segmented into activity segments using the motion segmentation based on the method in [23] with STIP histograms as the model observation. In addition to STIP feature, we collect two more features from these activity segments, namely Gist3D [7] and Action Bank [6].

STIP is a spatio-temporal local feature and widely used for representing human action in video. After collecting STIP features, an action video clip is represented by a histogram of BoW of these features [5]. On the other hand, Gist3D is a global video descriptor, which is computed by applying a bank of 3-D spatiotemporal filters on the frequency spectrum of a video sequence. Then dimensionality reduction methods can be applied to reduce the size of the feature vector. Unlike STIP and Gist3D, Action Bank is a higher level representation of human action, where an action clip is represented as the collected output of many template based action detectors. We select these three features from three different categories- low-level, global, and high-level respectively and use them separately to prove that our framework is independent of the type of feature used to represent an activity. We expect that the asymptotic performance of the framework would remain same for any other features.

3.2. Activity Model

We use an ensemble of multi-class linear Support Vector Machines (SVM) for activity modeling, which can be defined as follows: $\mathcal{H}(\mathbf{x}) = \sum_t \log \frac{1}{\beta_t} h_t(\mathbf{x})$, where h_t is the t^{th} classifier in the ensemble, $\beta_t = \epsilon_t/(1-\epsilon_t)$ is the corresponding weight, and ϵ_t is the normalized error due to h_t. A detailed mathematical analysis of ensemble of SVM classifiers can be found in [24].

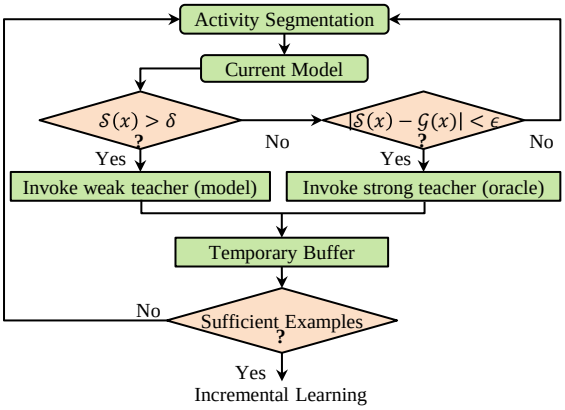

Figure 3: Flowchart of selecting examples for incremental learning and how to get the correct labels of these examples through active learning. $\mathcal{S}(\mathbf{x})$ and $\mathcal{G}(\mathbf{x})$ are defined in Section 3.3.

3.3. Active Learning System

According to [14], active learning can achieve greater learning accuracy with fewer training labels if the learner is allowed to choose the training data from which it learns. An active learner usually poses queries in the form of unlabeled training data instances to be labeled by an oracle. However, based on the type of teacher (oracle) available, the active learning system can be classified into two broad categories: strong teacher and weak teacher. Strong teachers are assumed to give correct and unambiguous class labels. Most, but not all, strong teachers are humans, which are assumed to have a significant cost. On the other hand, weak teachers generally provide more tentative labels. Most, but not all, weak teachers are assumed to be classification algorithms that make errors but perform above the accuracy of random guess [25]. Our proposed framework provides the opportunity to take advantages of both kind of teachers.

Active learning works within two common schemes: pool-based sampling and stream-based sampling [14]. In our proposed framework, we take the advantages of stream-based sampling, where unlabeled examples are presented one at a time and the learner must decide whether or not it is worth to invoke a teacher to label the example. Now, the following questions remain: When we should ask a teacher? Which teacher to invoke? And what action should we perform in response?

Teacher Selection: Details of the active learning mechanism are illustrated in Figure 3 using a flowchart. Whenever an unlabeled activity is presented to the system, the current activity recognition model is applied on the activity, which generates a tentative decision $\mathcal{H}(\mathbf{x})$, with a confidence score $\mathcal{S}(\mathbf{x})$. Let the second highest confidence score be $\mathcal{G}(\mathbf{x})$. $\mathcal{S}(\mathbf{x})$ and $\mathcal{G}(\mathbf{x})$ are defined as follows: $\mathcal{S}(\mathbf{x}) = \max_{y \in Y} \sum_k \sum_{t:\mathcal{H}_t(\mathbf{x})=y} \log \frac{1}{B_t}$, and $\mathcal{G}(\mathbf{x}) = \max_{y \in (Y - \mathcal{H}(\mathbf{x}))} \sum_k \sum_{t:\mathcal{H}_t(\mathbf{x})=y} \log \frac{1}{B_t}$, where $\mathbf{x} \in \mathbf{R}^n$ is the input activity, $y \in \{1, \ldots, Y\}$ are class labels, $\mathcal{H}_t(\mathbf{x})$ is a classifier, and $\log(1/B_t)$ is the corresponding weight. We invoke the weak teacher when the tentative decision $\mathcal{H}(\mathbf{x})$ has sufficiently large confidence score. That means, if $\mathcal{S}(\mathbf{x})$ is greater than a threshold δ, the unlabeled activity is labeled using the label $\mathcal{H}(\mathbf{x})$ from the current model. Else if, $|\mathcal{S}(\mathbf{x}) - \mathcal{G}(\mathbf{x})|$ is less than a threshold ϵ, the current model is not confident enough to decide about the label. This example lies near the decision boundary and possesses valuable information. In this case, the system invokes the strong teacher and obtains the label. Otherwise, the unlabeled activity is not used for incremental learning. When the system has accomplished the task of labeling the unlabeled activity, new activity $\mathbf{x}$ with label y is stored in a buffer temporarily. Choice of the parameters δ and ϵ are domain dependent and can be updated regularly based on system's performance. If the current model performs better on the unseen validation data, these parameters can be set such that the costly strong teacher is invoked rarely during training. Sensitivity analysis of these two parameters are provide in Section 4.

3.4. Incremental Activity Modeling

We present the detailed incremental activity modeling approach in Algorithm 1, while each of the steps is described in the following subsections.

Learning Prior Model: At first, we learn a prior model $\mathcal{H}_0$ using very few labeled training examples. In this work, we use an ensemble of SVMs as described in Section 3.2 as the prior model. Prior model learning stage is neither intensive like other state-of-the-art approaches, nor exhaustive in terms of covering all activity classes or in modeling the variation within the class. It is used as the starting point for the incremental learning.

Activity Segmentation and Active Learning: Let us consider that we have a video stream $\mathcal{V}$, starting at timestep t_0. As time progress, new activities are arriving from the streaming video. We segment an activity $\mathbf{x}_i$ at time $t_0 + i$ and collect features using the methods described in Section 3.1. We apply the current model $\mathcal{H}_k$ on the unlabeled activity $\mathbf{x}_i$ to get a label y_i using the active learning system described in Section 3.3. We store the labeled activity $(\mathbf{x}_i, y_i)$ temporarily in a buffer $\mathcal{B}_k$, where k stands for k^{th} incremental training step.

Algorithm 1: Incremental Activity Modeling.

Data: $\mathcal{V}$: Continuous Streaming Video.
Result: $\mathcal{H}$: Activity Recognition Model
$[(\mathbf{x}_{t_0+i}, y_{t_0+i})|i = 1, \ldots]$: Labeled Activities.
Parameters: Number of SVMs to be trained for batch k, T_k. Active learning parameters δ and ϵ.
Step 0: Learn the prior model $\mathcal{H}_0$ using fewer training data available.
Step 1: Segment the video $\mathcal{V}$ at timestamp $(t_0 + i)$ to get an unlabeled activity segment, $\mathbf{x}_i$ (Section 3.1).
Step 2: Apply the current model $\mathcal{H}_k$ on $\mathbf{x}_i$. Based on the condition met, get a label y_i for $\mathbf{x}_i$ (Section 3.3) and put $(\mathbf{x}_i, y_i)$ in the buffer, $\mathcal{B}_k$.
Step 3: If $\mathcal{B}_k$ contains m training examples, goto step 4 for next incremental learning, otherwise goto step 1.
Step 4: Initialize the distribution for selecting training examples: $\mathbf{w}_1(i) = D(i) = \frac{1}{m}, \forall i = \{1, \ldots, m\}$
Step 5:
for $t = 1$ *to* T_k **do**
1. Normalize distribution: $\mathbf{D}_t = \mathbf{w}_t / \sum_{i=1}^{m} \mathbf{w}_t(i)$
2. From $\mathcal{B}_k$, randomly choose 2/3 examples according to $\mathbf{D}_t$. Lets say them Tr_t.
3. Error: $\epsilon_t = 1$
4. while $\epsilon_t > 0.5$ **do**
Train a linear SVM, $h_t : \mathbf{x} \rightarrow y$ using Tr_t.
Error of h_t on $\mathcal{B}_k$, $\epsilon_t = \sum_{i:h_t(\mathbf{x}_i) \neq y_i} \mathbf{D}_t(i)$.
end
5. Normalized error: $\beta_t = \epsilon_t / (1 - \epsilon_t)$
6. Obtain the composite hypothesis and error:
$\mathcal{H}_t = \arg\max_{y \in Y} \sum_{t:h_t(\mathbf{x}_i)=y} \log(1/\beta_t)$
$E_t = \sum_{i:\mathcal{H}_t(\mathbf{x}_i) \neq y_i} \mathbf{D}_t(i) = \sum_{i}^{m} \mathbf{D}_t(i) |\mathcal{H}_t(\mathbf{x}_i) \neq y_i|$
7. If $E_t > 0.5$, set $t = t - 1$, discard $\mathcal{H}_t$ and goto line 2.
8. Normalized composite error, $B_t = E_t / (1 - E_t)$
9. Update the distribution of the training examples:
$\mathbf{w}_{t+1}(i) = \mathbf{w}_t(i) \times B_t^{1 - |H_t(\mathbf{x}_i) \neq y_i|}$
end
Step 6: Final decision on an unlabeled activity $\mathbf{x}$:
$\mathcal{H}(\mathbf{x}) = \arg\max_{y \in Y} \sum_k \sum_{t:\mathcal{H}_t(\mathbf{x})=y} \log \frac{1}{B_t}$
Step 7: Empty the buffer. Goto step 1 for incremental learning with next batch of training examples.

Incremental Learning: As in [12], our incremental learning approach is based on the following intuition: each new classifier added to the ensemble is trained using a set of examples drawn according to a distribution, which ensures that examples that are misclassified by the current ensemble have a high probability of being sampled in the next round. Weight update mechanism of the individual SVMs remains same as in [12], which we describe below.

Let us consider that inputs to the k^{th} incremental learning stage are a sequence of training examples, $\mathcal{B}_k$ =

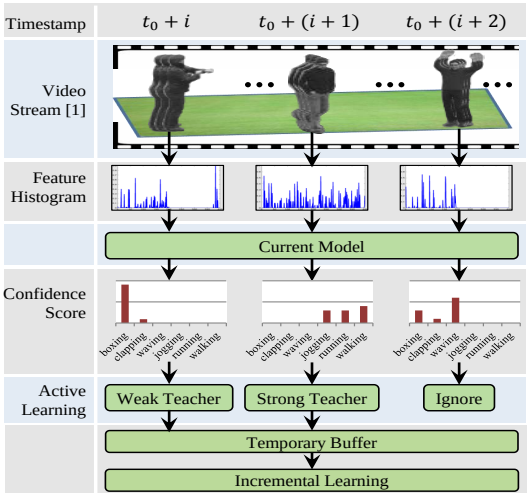

Figure 4: A sample run of our proposed incremental activity learning framework. After segmenting an activity from the video stream, we generate features and obtain a tentative label with a confidence score from the current model. Our active learning system analyzes the score and obtains the correct label for the activity by invoking a teacher. We temporarily store this new activity with the label for the next incremental learning step.

$\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$, where $\mathbf{x}_i$s are the training instances and y_is are the corresponding labels. Let us assume that a weak baseline SVM classifier model is known, and let T_k be the number of classifiers to be learned at the k^{th} stage. At first, training example distribution $\mathcal{D}$ is initialized by giving equal probability to all of the training examples. A subset of training examples Tr_t from $\mathcal{B}_k$ are selected according to the current distribution $\mathcal{D}_t$ at t^{th} classifier training step during k^{th} stage. A new weak baseline SVM classifier h_t is trained using these training examples. If the error associated with this new classifier, ϵ_t, is higher than a threshold 0.5, it will be rejected, otherwise it will be added to the ensemble $\mathcal{H}_t$. Then, error E_t of the ensemble $\mathcal{H}_t$ is computed on the training data. If the error associated with this updated ensemble $\mathcal{H}_t$ is higher than a threshold 0.5, the new update will be rejected and training h_t starts over again with new Tr_t. Training data distribution is updated at this point so that in the next round examples for which errors occurred get higher priority to be selected as the training example.

A sample run of our incremental learning framework on KTH dataset using STIP feature is illustrated in Figure 4. An activity is segmented at timestamp $t_0 + i$, which is followed by feature generation. New activity is labeled as "boxing" by the current model with a very high confidence score that leads the system to invoke the weak teacher. At timestamp $t_0 + (i + 1)$, current model labeled another new activity as "walking" with a lower confidence score, which leads the system to invoke the strong teacher. At timestamp $t_0 + (i + 2)$, the segmented activity is labeled as "waving", which is eventually ignored by the active learning system because the score is neither confident enough nor it is close to the decision boundary. Activities in the first two cases are stored temporarily in a buffer to be used as the training examples for the next incremental learning step.

4. Experiments

We conduct three experiments to verify the efficacy of our framework. Two of the experiments are carried out on the benchmark datasets KTH [1] and UCF11 [2], where we assume that activity segmentation is already done. We send the training examples as the unlabeled data to the incremental learning framework sequentially. We perform the third experiment on VIRAT ground human activity dataset [3], where we have to segment the activities from the video.

Objective: The main objective of these experiments is to analyze how well our framework incrementally learns the activity model with unlabeled data. We compare the performance of our framework (IL-unlabeled) with following three methods. Baseline-1: one time exhaustive learning with all of the available training examples. Baseline-2: one time exhaustive learning but with half of the available training examples, which are selected randomly. IL-labeled: our proposed incremental learning framework that uses only labeled examples.

Experiment Setup: We abide by the following rules during all experiments:

1. Due to the random selection of examples during training of SVM classifiers, each run of incremental learning on same dataset and features shows significant variance in accuracy. In order to get rid of this randomness, we average our results over multiple runs containing different random orders in which the data is presented.
2. For splitting training and test data, we perform five fold cross validation and then, average the results over these folds.

Presentation of Results: In the plots, we only show one accuracy (correct classification rate) over all of the activity classes. For example, KTH has six activity classes and the classification accuracy may be different for different activity classes. We average the results over all activity classes and show only this average accuracy in the plot due to space limitation. The x-axis in a plot illustrated in Figure 5 shows the fraction of training examples presented so far to the incremental learning methods, while the y-axis shows the classification accuracy on unknown test activities. Each of the results illustrated in Figure 5 was generated by us using the code provided by the original authors. There might be slight discrepancy between the shown results and the results that were claimed in the original paper. This is mainly due to the choices of different parameters during feature generation and classifier training. Parameters that were common across the different comparison methods were kept fixed as

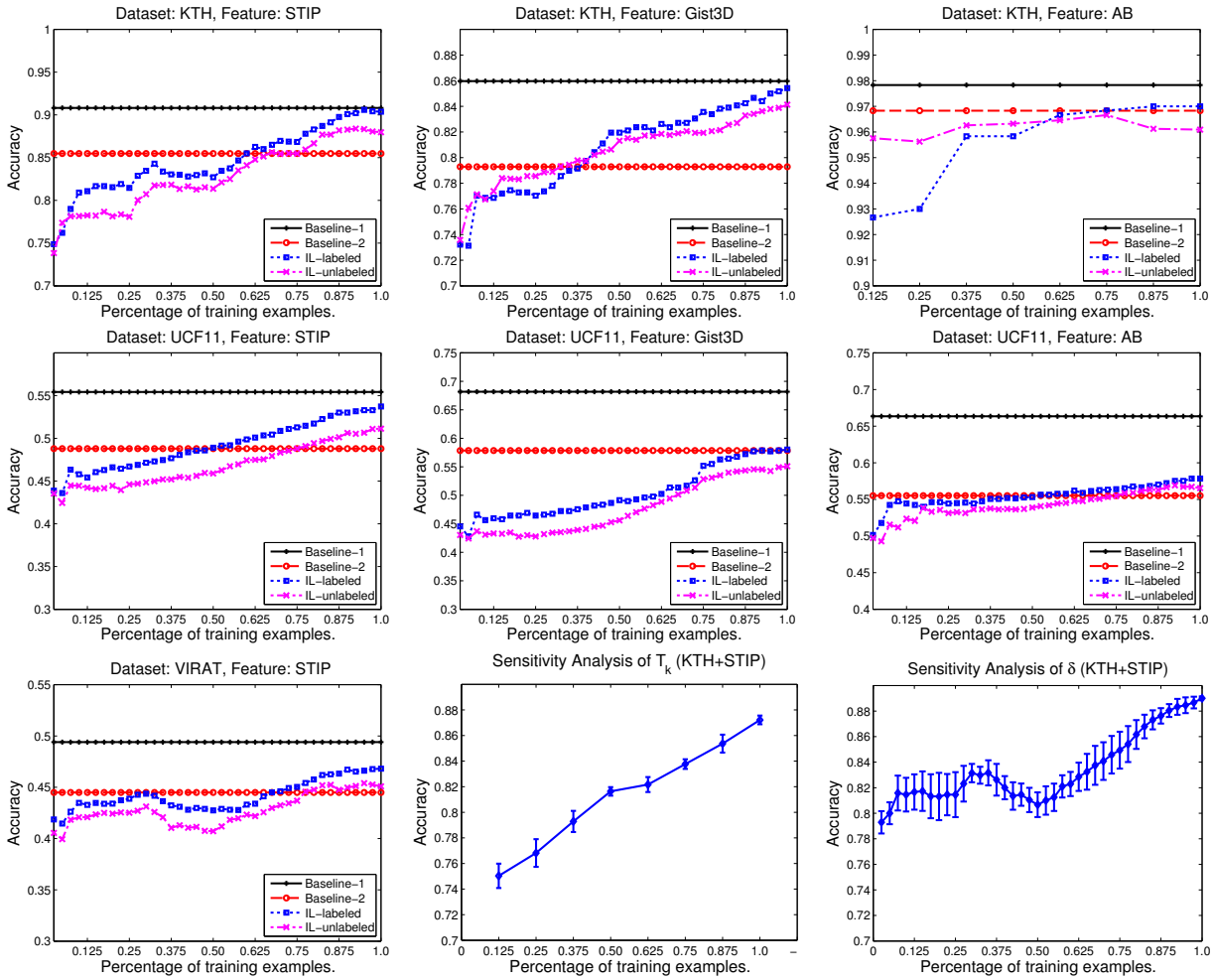

Figure 5: Top and Middle rows: performance comparison of our proposed method IL-unlabeled with Baseline-1, Baseline-2, and IL-labeled on KTH (top row) and UCF11 (middle row). Bottom row-left: performance on VIRAT. Bottom row-middle: sensitivity analysis of parameter T_k. Bottom row-right: sensitivity analysis of parameter δ. X-axis is the fraction of examples presented so far to the incremental learning framework and Y-axis is the accuracy of the classification. Plots are best viewable in color.

control variables. Additional results and analysis can be found in the supplementary material.

KTH Human Action Dataset: There are six actions in KTH [1] dataset: boxing, handclapping, handwaving, jogging, running and walking. These actions are performed by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors with lighting variation. There are totally 599 video clips with the resolution of 160×120 pixels.

Parameters in KTH Experiment: STIP: descriptor: HOG/HOF, descriptor size: 162, BoW dictionary size: 100, encoding type: 2x2 spatial, final feature vector size of an action clip: 400, $T_k = 10, \delta = 0.8, \epsilon = 0.2$. **Gist3D:** No. of clips: 3, size of feature vector: 104448, size of feature vector after PCA: 400, $T_k = 10, \delta = 0.7, \epsilon = 0.2$. **AB:** Size of feature vector: 14400, all other parameters remain same as in the original paper, $T_k = 2, \delta = 0.9, \epsilon = 0.3$.

UCF11 Human Action Dataset: The second experiment is performed on more challenging UCF11 dataset [2]. There are eleven actions in this dataset: basketball, biking, diving, golf_swing, horse_riding, soccer_juggling, swing, tennis_swing, trampoline_jumping, volleyball_spiking, and walking. These actions are performed by 25 subjects under different scenarios and illumination conditions. There are 1600 video clips with the resolution of 320×240 pixels.

Parameters in UCF11 Experiment: STIP: descriptor: HOG/HOF, descriptor size: 162, BoW dictionary size: 400, encoding type: 2x2 spatial, final feature vector size of an action clip: 1600, $T_k = 10, \delta = 0.5, \epsilon = 0.3$. **Gist3D:** No. of clips: 3, size of feature vector: 104448, size of feature vector after PCA: 1200, $T_k = 10, \delta = 0.6, \epsilon = 0.3$. **AB:** Size of feature vector: 14400, all other parameters remain same as in the original paper, $T_k = 10, \delta = 0.6, \epsilon = 0.3$.

VIRAT Human Activity Dataset: VIRAT Ground dataset [3] is a state-of-the-art streaming activity dataset with many challenging characteristics, such as wide variation in the activities and clutter in the scene. The dataset consists of surveillance videos of realistic scenes with different scales and resolution, each lasting 2 to 15 minutes and containing upto 30 events with 1920×1080 pixel resolution. The activities are 1. person loading an object to a vehicle; 2. person unloading an object from a vehicle; 3. person opening a vehicle trunk; 4. person closing a vehicle trunk; 5. person getting into a vehicle; 6. person getting out of a vehicle; 7. person gesturing; 8. person carrying an object; 9. person running; 10. person walking; 11. person entering a facility; and 12. person exiting a facility. We perform experiments using activities 1-6, 11, and 12 using only STIP feature. VIRAT is a new dataset, where existing methods are available only with STIP features and hence, we choose to restrict ourselves to this.

Parameters in VIRAT Experiment: STIP: descriptor type: HOG/HOF, descriptor vector size: 162, BoW dictionary size: 400, encoding type: None, final feature vector size for an action clip: 400, $T_k = 10, \delta = 0.6, \epsilon = 0.2$.

Results and Discussion: Results on the KTH dataset are shown in the top-row of Figure 5. It shows that incremental learning with labeled data using all of the three features- STIP, Gist3D, and AB- cross the accuracy of Baseline-2, while it touches the accuracy of Baseline-1 in case of STIP and Gist3D. The accuracy of incremental learning using unlabeled data is just below the accuracy of incremental learning using labeled data. Results on the UCF11 dataset are shown in the middle-row of Figure 5. It shows that incremental learning using features- STIP, Gist3D and AB with labeled data cross the accuracy of Baseline-2, while only STIP reaches near the accuracy of Baseline-1. With unlabeled data, STIP and AB cross the Baseline-2. Results on the VIRAT dataset are shown in the left plot of the bottom-row of Figure 5. It shows that incremental learning with labeled data and unlabeled data using STIP feature reach the accuracy between Baseline-1 and Baseline-2.

Above results demonstrate the effectiveness of the incremental learning framework- we achieve almost the same incremental learning accuracy as would have been the case if all the examples were previously labeled. Also, in most cases, we achieve close to the accuracy that would be obtained with a complete prior training phase. Exhaustive training in methods Baseline-1 and Baseline-2 require all of the examples to be available during training. On the other hand, incremental learning methods, labeled or unlabeled, do not require all of the previously seen data to be available. They only require to store a small amount of data for the current run of incremental learning. So, it would be unrealistic to expect that the maximum accuracy of the incremental learning methods would be better than the accuracy of Baseline-1. But we do expect that the accuracy of the incremental learning methods to be asymptotically increasing and at least better than the Baseline-2 method, which is exhibited in most of the cases.

Since, IL-labeled uses only the labeled data, the training accuracy is expected to be greater than the accuracy of IL-unlabeled. However, we see that the difference with the unlabeled case is very small, thus proving the efficiency of our proposed framework. Noticeably, IL-labeled and IL-unlabeled both perform better on KTH dataset than other two datasets by achieving performance closer to Baseline-1. The reason is that UCF11 and VIRAT are more challenging than KTH and it would require more examples to achieve the similar performance for these datasets. We show the performance of the proposed incremental activity learning framework on individual test action clips of different datasets in Figure 6.

Comparison with Other Methods: As discussed in Section 2, constraints of [19] (96.1%), [8] (90.3%) and [18] (96.3%) make them difficult to apply on streaming videos. Despite not assuming these constraints, our results (IL-labeled: 97%, IL-unlabeled: 96% using AB feature) are comparable or better on KTH (the only common dataset).

Percentage of Manual Labeling: Baseline-1 and Baseline-2 require manual labeling of 100% and 50% of all the data respectively. Also IL-labeled requires 100% manual labeling, although, data are presented to the system incrementally. However, in case of IL-unlabeled, about 15% and 25% data was needed to be manually labeled for KTH and UCF11 respectively. This shows that we can achieve close to state-of-the-art performance with far less tedious manual labor in labeling the examples.

Parameter Sensitivity: We analyze the sensitivity of two parameters T_k and δ on the system performance on KTH dataset using STIP features. Results of the sensitivity analysis of T_k (range 2 to 12) is shown in the middle plot of the bottom-row in Figure 5, while the sensitivity analysis of δ (range 0.7 to 0.9) is shown in the right plot of the bottom-row in Figure 5. Lower standard deviation of the accuracy proves that the choice of these parameters does not significantly effect the overall system performance.

5. Conclusion and Future Works

In this work, we proposed a framework for incremental activity modeling. Our framework took advantage of state-of-the-art machine learning tools and active learning to learn activity models incrementally over time. We performed detailed experiments on multiple challenging datasets. Results show the robustness of our approach as accuracy asymptotically increases in all of the cases. Future works will investigate new tools and techniques so that we can incrementally learn unseen activity classes as well. We will also improve our framework so that we can model

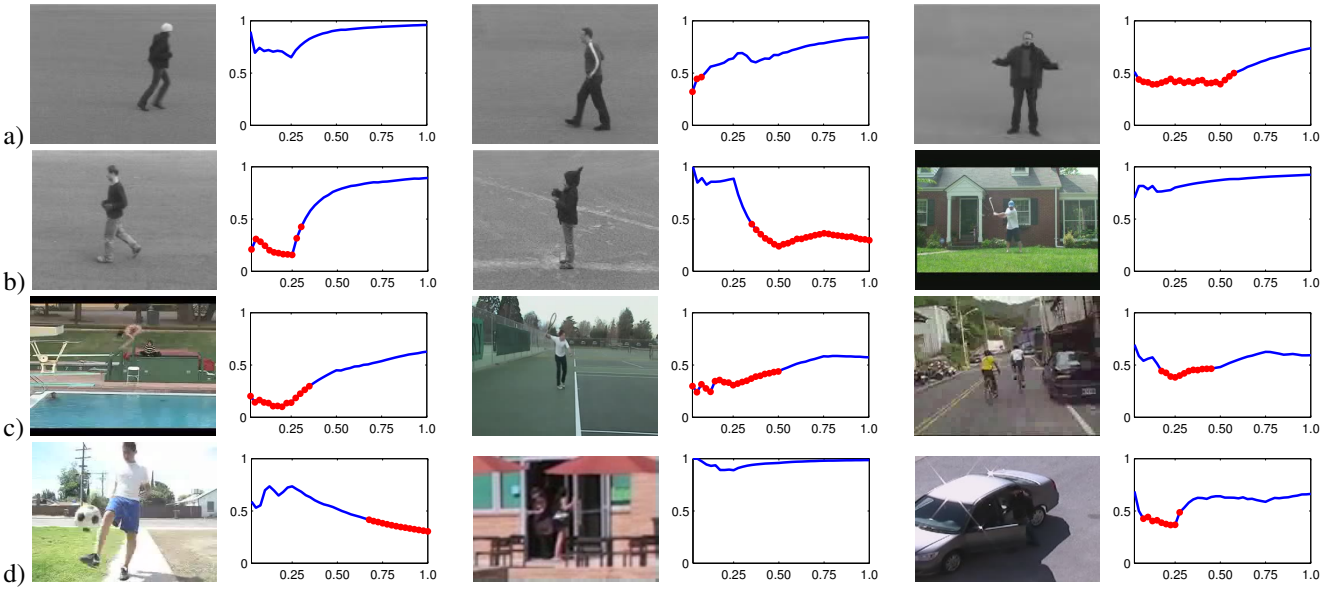

Figure 6: This figure shows the performance of the proposed incremental activity modeling framework on individual test action clips of different datasets. Above illustrated actions are as follows (left to right, top to bottom): a) KTH:jogging, walking, handclapping; b) running, boxing, UCF11:golf_swing; c) diving, tennis_swing, biking; d) socer_juggling, VIRAT:facility_out, and vehicle_in. X-axis is the fraction of the examples presented so far to the incremental learning framework and Y-axis is the normalized confidence score $\mathcal{H}(\mathbf{x})$. Blue line means correct classification of the action, while red spots means missclassificaiton of the action at that particular instant. In most of the cases, our proposed incremental learning framework increases the confidence score of an action and can retain the correct classification; in some cases, updated model rectifies the missclassifications (red to blue). In some rare cases (b- KTH:boxing and d- UCF11:socer_juggling), our framework failed to perform well and missclassified an action even though it was correctly classified before (blue to red). Plots are best viewable in color.

more complex activities and update them in real time.

References

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *ICPR*, 2004.

[2] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *CVPR*, 2009.

[3] S. Oh, A. Hoogs, and et. al., "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011.

[4] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, 2010.

[5] I. Laptev, "On space-time interest points," *IJCV*, 2005.

[6] S. Sadanand and J. Corso, "Action bank: A high-level representation of activity in video," in *CVPR*, 2012.

[7] B. Solmaz, S. M. Assari, and M. Shah, "Classifying web videos using a global video descriptor," *MVAP*, 2012.

[8] R. Minhas, A. Mohammed, and Q. Wu, "Incremental learning in human action recognition based on snippets," *IEEE TCSVT*, 2012.

[9] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *ICCV*, 2011.

[10] Z. Si, M. Pei, B. Yao, and S.-C. Zhu, "Unsupervised learning of event and-or grammar and semantics from video," in *ICCV*, 2011.

[11] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware modeling and recognition of activities in video," in *CVPR*, 2013.

[12] R. Polikar, L. Upda, S. Upda, and V. Honavar, "Learn++: an incremental learning algorithm for supervised neural networks," *IEEE TSMC Part:C*, vol. 31, no. 4, 2001.

[13] H. He, S. Chen, K. Li, and X. Xu, "Incremental learning from stream data," *IEEE TNN*, vol. 22, no. 12, pp. 1901–1914, 2011.

[14] B. Settles, "Active learning," *Morgan & Claypool*, 2012.

[15] J. M. Buhmann, A. Vezhnevets, and V. Ferrari, "Active learning for semantic segmentation with expected change," in *CVPR*, 2012.

[16] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell, "Active learning with gaussian processes for object categorization," in *ICCV*, 2007.

[17] C. Loy, T. Xiang, and S. Gong, "Stream-based active unusual event detection," in *ACCV*, 2011.

[18] X. Liu and J. Zhang, "Active learning for human action recognition with gaussian processes," in *ICIP*, 2011.

[19] K. Reddy, J. Liu, and M. Shah, "Incremental action recognition using feature-tree," in *ICCV*, 2009.

[20] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *ICPR*, 2004.

[21] P. F. Felzenszwalb, R. B. Girshic, and D. McAllester. Discriminatively trained deformable part models, release 4. [Online]. Available: http://people.cs.uchicago.edu/pff/latent-release4/

[22] B. Song, T. Jeng, E. Staudt, and A. Roy-Chowdury, "A stochastic graph evolution framework for robust multi-target tracking," in *ECCV*, 2010.

[23] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal., "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *CVPR*, 2009.

[24] R. Schapire, "Strength of weak learning," *ML*, 1990.

[25] Y. Hao, Y. Chen, J. Zakaria, B. Hu, T. Rakthanmanon, and E. Keogh, "Towards never-ending learning from time series streams," in *SIGKDD*, 2013.