Inferring Temporal Compositions of Actions Using Probabilistic Automata

Rodrigo Santa Cruz[1,2,*], Anoop Cherian[3], Basura Fernando[4], Dylan Campbell[2], and Stephen Gould[2]
[1]The Australian e-Health Research Centre, CSIRO, Brisbane, Australia
[2]Australian Centre for Robotic Vision (ACRV), Australian National University, Canberra, Australia
[3]Mitsubishi Electric Research Labs (MERL), Cambridge, MA
[4]A*AI, A*STAR Singapore

Abstract

This paper presents a framework to recognize temporal compositions of atomic actions in videos. Specifically, we propose to express temporal compositions of actions as semantic regular expressions and derive an inference framework using probabilistic automata to recognize complex actions as satisfying these expressions on the input video features. Our approach is different from existing works that either predict long-range complex activities as unordered sets of atomic actions, or retrieve videos using natural language sentences. Instead, the proposed approach allows recognizing complex fine-grained activities using only pre-trained action classifiers, without requiring any additional data, annotations or neural network training. To evaluate the potential of our approach, we provide experiments on synthetic datasets and challenging real action recognition datasets, such as MultiTHUMOS and Charades. We conclude that the proposed approach can extend state-of-the-art primitive action classifiers to vastly more complex activities without large performance degradation.

1. Introduction

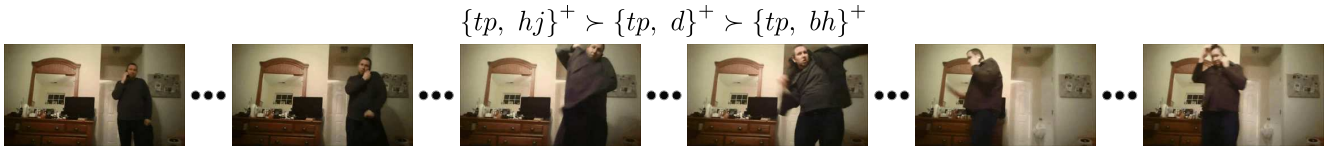
Real-world human activities are often complex combinations of various simple actions. In this paper, we define compositional action recognition as the task of recognizing complex activities expressed as temporally-ordered compositions of atomic actions in videos. To illustrate our task, let us consider the video sequence depicted in Figure 1. Our goal is to have this video clip retrieved from a large collection of videos. A natural query in this regard can be: *"find videos in which someone is holding a jacket, dressing, and brushing hair, while talking on the phone?"*. As is clear, this query combines multiple atomic actions such as "holding a jacket", "dressing", etc.; however, we are interested only in videos that adhere to the temporal order provided in the query. Such a task is indispensable in a variety of real-world applications, including video surveillance [2], patient monitoring [17], and shopping behavior analysis [19].

The current state-of-the-art action recognition models are usually not equipped to tackle this compositional recognition task. Specifically, these models are often trained to either discriminate between atomic actions in a multi-class recognition setting – such as "holding a jacket", "talking on the phone", etc. [1, 32], or generate a set of action labels – as in a multi-label classification setup – for a given clip, e.g., {"cooking a meal", "preparing a coffee"} [9, 10, 34]. On the one hand, extending the multi-class framework to complex recognition may need labels for all possible compositions – the annotation efforts for which could be enormous. On the other hand, the multi-label formulation is designed to be invariant to the order of atomic actions and thus is sub-optimal for characterizing complex activities.

Recent methods attempt to circumvent these limitations by leveraging textual data, allowing zero-shot action recognition from structured queries expressed as natural language sentences. For instance, these are models able to perform zero-shot action classification [13, 23], action localization [3, 6, 20], and actor-action segmentation [4] from natural language sentences. However, such descriptions are inherently ambiguous and not suitable for precisely describing complex activities as temporal compositions of atomic actions. As an example, all natural language descriptions listed for the video shown in Figure 1 are true, but none of them describe the sequence of events precisely.

Building on the insight that complex activities are fundamentally compositional action patterns, we introduce a probabilistic inference framework to *unambiguously* describe and efficiently recognize such activities in videos. To this end, we propose to describe complex activities as regular expressions of primitive actions using *regular language operators*. Then, using probabilistic automata [26], we develop an inference model that can recognize these regular expressions in videos. Returning to the example in Figure 1, given the primitive actions "talking on the

*Corresponding author. Email: rodrigo.santacruz@csiro.au



Natural language queries: **(a)** "Someone is talking on the phone, dressing a jacket and brushing hair."; **(b)** "Someone is talking on the phone and holding a jacket, then he dresses it and brushes his hair."; **(c)** "Someone is talking on the phone while dressing a jacket and brushing hair.";

Figure 1. A complex activity can be described by natural language queries, which are often incomplete and have vague and/or ambiguous temporal relations between the constituent actions. For instance, option (a) does not mention all the actions involved, and it is not clear from options (b) and (c) whether the actions happen simultaneously or sequentially. In contrast, a regular expression of primitive actions can precisely describe the activity of interest. For instance, given the primitive actions "talking on the phone" (tp), "holding a jacket" (hj), "dressing" (d), and "brushing hair" (bh), the regular expression $\{tp,\ hj\}^+ \succ \{tp,\ d\}^+ \succ \{tp,\ bh\}^+$ precisely describes the activity depicted in the frames, where the sets of primitive actions ($\{\cdot\}$), the regular language operator 'concatenation' ($\succ$) and the operator 'one-or-more repetition' ($+$) define concurrent, sequential and recurrent actions, respectively.

phone" (tp), "holding a jacket" (hj), "dressing" (d), and "brushing hair" (bh), the regular expression $\{tp,\ hj\}^+ \succ \{tp,\ d\}^+ \succ \{tp,\ bh\}^+$ precisely describes the activity in the video, where the sets of primitives ($\{\cdot\}$), the regular language operators 'concatenation' ($\succ$), and 'one-or-more repetition' operator ($+$) define concurrent, sequential, and recurrent actions respectively.

Our framework offers some unique benefits. It can recognize rare events, such as an "Olympic goal" for example, by composing "corner kick", "ball traveling", and "goal". Further, it does not require any additional data or parameter learning beyond what is used for the primitive actions. Thus, it can scale an existing pretrained action recognition setup towards complex and dynamic scenarios.

Below we summarize our main contributions:

1. We propose to recognize complex activities as temporal compositions of primitive action patterns. We formulate a framework for this task that resembles a regular expression engine in which we can perform inference for any compositional activity that can be described as a regular expression of primitives.
2. Using probabilistic automata, we derive a model to solve the inference problem leveraging pretrained primitive action classifiers, without requiring additional data, annotations or parameter learning.
3. Apart from experiments on a synthetic dataset, we evaluate the proposed framework on the previously described task of compositional action recognition using trimmed and untrimmed videos from challenging benchmarks such as MultiTHUMOS [35] and Charades [30].

2. Related Work

The action recognition community has mostly focused on developing models to discriminate between short and simple actions [7, 15]. While getting very accurate in this context, these models require training data for every action of interest which is often infeasible. Zero-shot learning approaches [5, 13, 16, 21] were developed to mitigate this problem, but these models still interpret action recognition as the assignment of simple action labels. In contrast, we follow a compositional view of action recognition where complex actions are inferred from simple primitive actions.

Recently, the community has shifted focus to the recognition of the long-tailed distribution of complex activities. However, existing methods [9, 10, 34] tackle this problem in a multi-label classification setup where the goal is to predict the atomic actions necessary to accomplish a complex activity under weak and implicitly-defined temporal relationships. In contrast, our proposed framework allows querying complex activities with specific and explicitly provided temporal order on the atomic actions.

Another approach to recognize complex activities with partial temporal structure is to leverage vision-language models [3, 4, 6, 20]. We argue, however, that natural language sentences may lead to ambiguous descriptions of complex activities as shown in Figure 1, which makes it difficult to ground the videos and their queries. Instead, we resort to regular languages that are designed to unambiguously describe concepts and also allows efficient inference.

Serving as inspirations for our approach, the works of İkizler and Forsyth [11] and Vo and Bobick [33] recognize human-centered activities from compositions of primitive actions. However, these approaches can only query for sequential or alternative primitive actions of *fixed length*. In contrast, we propose a more expressive and complete language for querying complex activities allowing sequential, concurrent, alternative, and recursive actions of varying lengths. Furthermore, our work focuses on zero-shot recognition of complex activities, unlike these approaches which require training data for the queried activities.

Note that our work is different from ones that leverage manually-annotated training data to perform structured predictions involving primitive actions. For instance, Richard and Gall [28] and Piergiovanni and Ryoo [25] consistently label video frames in a sequence, while Ji et al.'s model [14] outputs space-time scene graphs capturing action-object interactions. Differently, we tackle zero-shot activity classification over a highly complex label space (*i.e.*, the space

of all regular expressions of primitive actions) by using a probabilistic inference framework that uses only pretrained primitive action classifiers and does not need training data for action composites, or classifier finetuning.

3. Approach

In this section, we first formalize the problem of recognizing complex activities described by regular expressions of primitive actions. Then, we derive a deterministic and probabilistic inference procedure for this problem.

3.1. Problem Formulation

Initially, let us assume the existence of a pre-defined set of known actions, called primitives. For example, "driving" (d), "getting into a car" (gc), "talking on the cellphone" (tc), and "talking to someone" (ts). These primitives can also happen simultaneously in time, which we express as subsets of these primitive actions, *e.g.*, $\{a_d,\ a_{tc}\}$ means someone driving and talking on the cellphone at the same time. Moreover, consider three basic composition rules inspired by standard regular expression operators: concatenation ($\succ$), alternation ($|$), and Kleene star ($\star$) denoting sequences, unions and recurrent action patterns, respectively. Note also that more complex operators can be defined in terms of these ones, *e.g.*, one-or-more repetition of action ($+$) is defined as $a^+ \triangleq a \succ a^\star$. Then, from any complex activity described as a composition of subsets of primitive actions and these operators, our goal is to recognize whether a given video depicts the described activity. For instance, whether a given YouTube video depicts the complex activity "someone driving and talking on the phone or talking to someone, repeatedly, just after getting into a car", which can be described unambiguously as "$a_{gc} \succ (\{a_d,\ a_{tc}\} \,|\, \{a_d,\ a_{ts}\})^+$".

Formally, let us define a set of *primitive actions* $\mathcal{A} = \{a_i\}_{i=1}^{M}$. We can express a complex activity by forming *action patterns*, an arbitrary regular expression r combining subsets of primitives $w \in \mathcal{P}(\mathcal{A})$, where $\mathcal{P}(\mathcal{A})$ is the power-set of $\mathcal{A}$, with the aforementioned *composition rules* $\mathcal{O} = \{\succ, |, \star\}$. Note that background actions and non-action video segments are represented by the null primitive $\emptyset \in \mathcal{P}(\mathcal{A})$. Our goal then is to model a function $f_r : \mathcal{V} \to [0, 1]$ that assigns high values to a video $v \in \mathcal{V}$ if it depicts the action pattern described by the regular expression r and low values otherwise.

This work focuses on solving the aforementioned problem by developing a robust inference procedure leveraging state-of-the-art pretrained primitive action classifiers. Such an approach does not require extra data collection, parameter learning or classifier finetuning. Indeed, learning approaches are beyond the scope of this paper and a compelling direction for future work.

3.2. Deterministic Model

Regular expressions are used to concisely specify patterns of characters for matching and searching in large texts [18, 24, 29]. Inspired by this idea, we now describe a deterministic model based on deterministic finite automata (DFA) [22, 27] to the problem of recognizing action patterns in videos.

Let us start by defining a DFA M_r for an action pattern r as a 5-tuple $(\mathcal{Q}, \Sigma, \delta, q_0, \mathcal{F})$, consisting of a finite set of states Q, a finite set of input symbols called the alphabet Σ, a transition function $\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q}$, an initial state $q_0 \in \mathcal{Q}$, and a set of accept states $\mathcal{F} \subseteq \mathcal{Q}$. In our problem, the alphabet Σ is the power-set of action primitives $\mathcal{P}(\mathcal{A})$ and the transition function δ is a lookup table mapping from a state $q_i \in \mathcal{Q}$ and a subset of primitives $w \in \Sigma$ to another state $q_j \in \mathcal{Q}$ or halting the automaton operation if no transition is defined. Note that all these structures are efficiently constructed and optimized from a given action pattern r using traditional algorithms such as non-deterministic finite automaton (NFA) construction [12], the NFA to DFA subset construction [27], and Hopcroft's DFA minimization [8].

Additionally, let us denote the subset of primitive actions happening in a given frame x as $w(x) = \{a \in \mathcal{A} \mid p(a|x) \geq \tau\}$, where $p(a|x)$ is the probability of a primitive action $a \in \mathcal{A}$ happening in frame x and $\tau \in [0, 1]$ is a threshold hyper-parameter. In this formulation, $p(a|x)$ can be built from any probabilistic action classifier, while τ should be set by cross-validation. Then, we say that the deterministic model accepts an input video $v = \langle x_1, \ldots, x_n \rangle$ if and only if there exists a sequence of states $\langle q_0, \ldots, q_n \rangle$ for $q_i \in \mathcal{Q}$ such that (i) the sequence starts in the initial state q_0, (ii) subsequent states q_i satisfy $q_i = \delta(q_{i-1}, w(x_i))$ for $i = 1, \ldots, n$, and (iii) the sequence finishes in a final state $q_n \in \mathcal{F}$.

This procedure defines a binary function that assigns a value of one to videos that reach the final state of the compiled DFA M_r and zero otherwise. This is a very strict classification rule since a positive match using imperfect classifiers is very improbable. In order to relax such a classification rule, we propose implementing the score function

$$f_r(v) = \frac{\operatorname{dist}(q_0, \hat{q})}{\operatorname{dist}(q_0, \hat{q}) + \min_{q_f \in \mathcal{F}} \operatorname{dist}(\hat{q}, q_f)}\,, \qquad (1)$$

where $\hat{q}$ is the state in which the compiled DFA M_r halted when simulating the sequence of frames defined by the video v, and the function $\operatorname{dist}(q_x, q_y)$ computes the minimum number of transitions to be taken to reach the state q_y from state q_x. That is, for a given regular expression, the deterministic model scores a video according to the fraction of transitions taken before halting in the shortest path to a final state in the compiled DFA.

In short, the deterministic model implements the function f_r by computing Equation 1 after simulating the DFA

M_r compiled for the regular expression r on the sequence of subsets of primitive actions $w(x)$ generated by thresholding the primitive action classifiers $p(a|x)$ on every frame x of the input video v.

3.3. Probabilistic Model

In order to take into account the uncertainty of the primitive action classifiers' predictions, we now derive a probabilistic inference model for our problem. Specifically, we propose to use Probabilistic Automata (PA) [26] instead of DFAs as the backbone of our framework.

Mathematically, let us define a PA U_r for a regular expression r as a 5-tuple $(\mathcal{Q}, \Sigma, T, \boldsymbol{\rho}, \mathcal{F})$. $\mathcal{Q}$, Σ, and $\mathcal{F}$ are the set of states, the alphabet, and the final states, respectively. They are defined as in the deterministic case, but an explicit reject state is added to $\mathcal{Q}$ in order to model the halting of an automaton when an unexpected symbol appears in any given state. $T = \{\forall w \in \Sigma : T_w \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{Q}|}\}$ is the set of row stochastic transition matrices T_w associated with the subset of primitives $w \in \Sigma$ in which the entry $[T_w]_{i,j}$ is the probability that the automaton transits from the state q_i to the state q_j after seeing the subset of primitives w. Likewise, $\boldsymbol{\rho} \in \mathbb{R}^{|\mathcal{Q}|}$ is a stochastic vector and $[\boldsymbol{\rho}]_i$ is the probability that the automaton starts at state q_i.

Note that all these structures are estimated from the transition function δ and initial state q_0 of the compiled DFA M_r for the same regular expression r as follows,

$$
\begin{aligned}
[T_w]_{i,j} &= \frac{[\![\delta(i,w) = j]\!] + \alpha}{\sum_{k\in\mathcal{Q}} [\![\delta(i,w) = k]\!] + \alpha|\mathcal{Q}|}\,, \\
[\boldsymbol{\rho}]_i &= \frac{[\![q_0 = i]\!] + \alpha}{\sum_{k\in\mathcal{Q}} [\![q_0 = k]\!] + \alpha|\mathcal{Q}|}\,,
\end{aligned}
\tag{2}
$$

where the indicator function $[\![c]\!]$ evaluates to one when the condition c is true and zero otherwise. The smoothing factor α is a model hyper-parameter that regularizes our model by providing non-zero probability for every distribution in our model. As mentioned before, a hypothetical dataset of action patterns and videos pairs could be leveraged by a learning algorithm to fit these distributions, but the current work focuses on the practical scenario where such a training set is difficult to obtain.

However, PAs do not model uncertainty in the input sequence which is a requirement of our problem, since we do not know what actions are depicted in a given video frame. Therefore, we propose to extend the PA framework by introducing a distribution over the alphabet given a video frame. In order to make use of off-the-shelf action classifiers like modern deep leaning models, we assume independence between the primitive actions and estimate the probability of a subset of primitives given a frame as

$$
p(w|x) = \left(\prod_{a\in\mathcal{A}} p(a|x)^{[\![a\in w]\!]} \left(1 - p(a|x)\right)^{[\![a\notin w]\!]} \right)^{\gamma}, \tag{3}
$$

where $p(a|x)$ is the prediction of a primitive action classifier as before and γ is a hyper-parameter that compensates for violations to the independence assumption. After such a correction, we need to re-normalize the $p(w|x)$ probabilities in order to form a distribution.

Making use of this distribution, we derive the induced (row stochastic) transition matrix $I(x) \in \mathbb{R}^{|\mathcal{Q}|\times|\mathcal{Q}|}$ after observing a video frame x by marginalizing over the alphabet Σ as follows,

$$
I(x) = \sum_{w\in\Sigma} T_w p(w \mid x), \tag{4}
$$

where the entry $[I(x)]_{i,j}$ denotes the probability of transiting from state q_i to state q_j after seeing a frame x. It is also important to note that naively computing this induced transition matrix is problematic due to the possibly large alphabet Σ. For instance, a modestly sized set of 100 primitive actions would generate an alphabet of 2^{100} subsets of primitives. In order to circumvent such a limitation, we factorize Equation 4 as

$$
I(x) = \sum_{w\in\Sigma'} T_w p(w|x) + \bar{T}\Big(1 - \sum_{w\in\Sigma'} p(w|x)\Big), \tag{5}
$$

where we first define a typically small subset of our alphabet $\Sigma' \subseteq \Sigma$ composed of subsets of primitives that have at least one transition in the compiled DFA M_r. Then, we make use of the fact that the remaining subsets of primitives $\Sigma \setminus \Sigma'$ will be associated with exactly the same transition matrix, denoted by $\bar{T}$ and also computed according to Equation 2, and the sum of their probability in a given frame is equal to $1 - \sum_{w\in\Sigma'} p(w \mid x_i)$. Therefore, Equation 5 computes the induced transition matrix efficiently, without enumerating all subsets of primitives in the alphabet.

Finally, we can compute the normalized matching probability between a video $v = \langle x_1, \dots x_n \rangle$ and the regular expression r as the probability of reaching a final state in the compiled PA U_r as

$$
P_{U_r}(v) = \left(\boldsymbol{\rho}^{\intercal} \prod_{i=1}^{|v|} I(x_i) \right)^{\frac{1}{|v|}} \boldsymbol{f}\,, \tag{6}
$$

where $\boldsymbol{f}$ is an indicator vector such that $\boldsymbol{f}_i = 1$ if and only if $q_i \in \mathcal{F}$ and 0 otherwise. The normalization by $1/|v|$ calibrates the probabilities to allow comparisons between videos of different length.

Intuitively, the proposed probabilistic inference model implements the function f_r by first converting the compiled

DFA M_r, for the regular expression r, to a PA U_r according to Equation 2. Then, as described in Equation 6, this model keeps a distribution over the states $\mathcal{Q}$ starting from the initial state distribution $\boldsymbol{\rho}$ and updating it according to the induced transition matrix $I(x)$, defined in Equation 5, as we observe the input video frames x. Finally, as also described in Equation 6, the matching probability is computed as the sum of the probability in the final states once all of the input video frames are observed.

4. Experiments

We now evaluate the proposed inference models for rich compositional activity recognition. We first perform a detailed analysis of the proposed approaches on controlled experiments using synthetic data. Then, we test the utility of our methods on challenging action recognition tasks.

4.1. Synthetic Analysis

It is unrealistic to collect video data for the immense number of possible regular expressions that our models may encounter. As such, we resort to the use of synthetically generated data inspired by the well known Moving MNIST dataset [31]. More specifically, we develop a parametrized data generation procedure to produce moving MNIST videos depicting different patterns of appearing MNIST digits. Such a procedure can generate videos that match regular expressions of the form

$$w_1^+ \succ \cdots \succ \left(\left(w_s^{1+} \succ \cdots \succ w_n^{1+} \right) \middle| \cdots \middle| \left(w_s^{d+} \succ \cdots \succ w_n^{d+} \right) \right), \quad (7)$$

where the symbols $w \in \mathcal{P}(\mathcal{A})$ are subsets of the primitives $\mathcal{A}$ which are the ten digit classes. The data generation procedure has the following parameters: the number of primitives that simultaneously appear in a frame $|w|$, the total number of different sequential symbols n, the number of alternative sequences of symbols d, the start position s of each alternative sequence in the pattern, and the total number of generated frames. Since complex patterns can match different sequences of symbols due to the the alternation operator ($|$), we perform random walks from the start state until reaching a final state in the compiled DFA in order to generate video samples for a given regular expression. Figure 2 presents an example of regular expressions and video clips generated by this data generation procedure.

Using the synthetically generated data, we first train the primitive classifiers on frames depicting a different number of digits obtained from the MNIST training split. The primitive classifiers consist of a shallow CNN trained to minimize the binary cross entropy loss for all digits in a vast number of frames. In order to evaluate the robustness of the proposed models, we also generate worse versions of these classifiers by adding noise to their predictions. Figure 2 shows the performance of the learned primitive classifiers

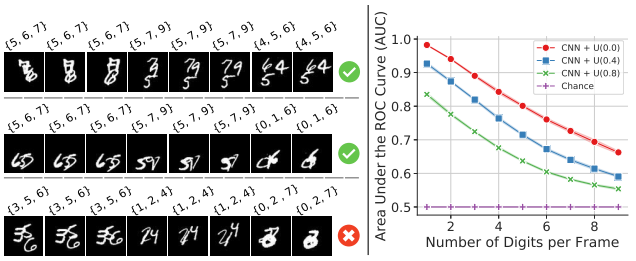


Figure 2. **Left:** Video samples synthetically generated for the regular expression "$\{a_5,\, a_6,\, a_7\}^+ \succ \{a_5,\, a_7,\, a_9\}^+ \succ \left(\{a_4,\, a_5,\, a_6\}^+ \,|\, \{a_0,\, a_1,\, a_6\}^+\right)$" which has $|w| = 3$ digits per frame, $n = 3$ different sequential symbols, $d = 2$ alternative sequences starting from $s = 2$, depicted in a total of 8 frames. The two first rows are clips that match the given regular expression, while the last row depicts a negative video clip. **Right:** The performance of the primitive classifiers on the test set with a different number of digits per frame and under different noise levels. $U(x)$ denotes uniform additive noise between $[-x, x]$ and the classifiers' predictions are re-normalized using a softmax function.

on different levels of noise and different numbers of digits per frame. Note that more digits per frame implies more occlusion between digits since the frame size is kept constant, which also decreases the classifier's performance.

Finally, using this synthetic data and the trained primitive classifiers, we test our models for the inference of different regular expressions by setting all the data generation parameters to default values with the exception of the one being evaluated. We use the values described in Figure 2 as default values, but we generate video clips of 32 frames. In Figure 3, we plot standard classification/retrieval metrics, e.g., Area Under the ROC Curve (AUC) and Mean Average Precision (MAP), against different data generation parameters. More specifically, at each configuration, using the MNIST test split, we generate 100 expressions with 20 positive samples, totaling about 2000 video clips. In order to robustly report our results, we repeat the experiment ten times reporting the mean and standard deviation of the evaluation metrics. We also cross-validate the model hyperparameters, τ for the deterministic model and α and γ for the probabilistic model, in a validation set formed by expressions of similar type as the ones to be tested, but with video clips generated from a held-out set of digit images extracted from the training split of the MNIST dataset.

As can be seen, the probabilistic model performs consistently better than the deterministic model in all experiments, providing precise and robust predictions. Furthermore, the probabilistic model is more robust to high levels of noise in the primitive classifiers' predictions. While the deterministic model works as poorly as random guessing with high noise levels, *e.g.* $U(0.8)$, the probabilistic model still produces good results. The probabilistic model also works consistently across different kinds of regular expressions. Indeed, its performance is almost invariant to

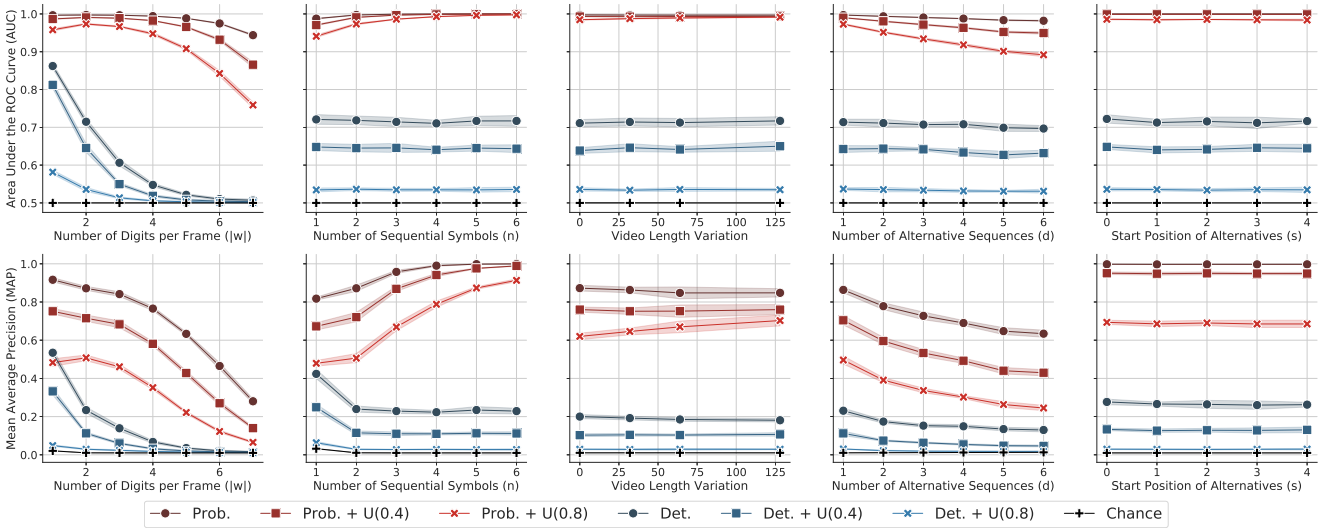


Figure 3. Plots of the performance, in terms of AUC and MAP, of the proposed methods on the generated synthetic dataset using primitive classifiers with different levels of noise as shown in Figure 2. The generated data consists of video clips depicting regular expressions parametrized according to Equation 7. We evaluate the proposed approaches according to the following data parameters: the number of digits that simultaneously appear in a frame $|w|$, the total number of different sequential symbols n, the variance in number of frames in the videos, the number of alternative sequences of symbols d, and the start position s of each alternative sequence in the pattern respectively.

most of the regular expressions parameters evaluated. In the case of the number of digits per frame $|w|$ for which relevant performance degradation is observed, the performance degradation correlates with the decrease in performance presented by the primitive classifiers as the number of digits per frames is increased (see Figure 2). The probabilistic model, however, is able to mitigate this degradation. For example, comparing the performance at two and five digits per frame, we observe that a drop of about 16% in AUC on the primitive classifiers performance causes a reduction smaller than 3% in AUC for the probabilistic model.

4.2. Action Recognition

We now evaluate our models on action recognition problems. We first describe the experimental setup, metrics and datasets used in our experiments. Then we analyze how effectively our model can recognize activities described by regular expressions in trimmed and untrimmed videos.

Experimental Setup. In order to evaluate the proposed inference models in the action recognition context, we collect test datasets of regular expressions and video clips by mining the ground-truth annotation of multilabel action recognition datasets such as Charades [30] and MultiTHUMOS [35]. More specifically, we search for regular expressions of the type defined in Equation 7 where the symbols w are subsets of the primitive actions annotated in the datasets. Charades has 157 actions, while MultiTHUMOS has 65 actions. Given the regular expression parameters, we first form instances of regular expressions using the primitive actions present in the datasets, keeping the ones that have at least one positive video clip. Then, using these instances of regular expressions, we search for all positive video clips in the dataset in order to form a new dataset of regular expressions and video clips which will be used in our experiments.

As primitive action classifiers, we use the I3D model proposed by Carreira and Zisserman [1] pretrained on the Kinetics dataset and finetuned on the training split of the Charades and MultiTHUMOS datasets to independently recognize the primitive actions. In this work, we only use the I3D-RGB stream, but optical flow and other information can be easily added since our formulation depends only on the final predictions of the primitive classifiers. Using the frame-level evaluation protocol (*i.e.* Charades localization setting), these classifiers reach **16.12% and 24.93% in MAP** on classifying frames into primitive actions on the test split of Charades and MultiTHUMOS datasets respectively.

Once the primitives classifiers are defined, we setup the deterministic and probabilistic inference models with them, cross-validate these inference model hyper-parameters using expressions and video clips mined from the training split, and evaluate these inference models in a different set of expressions mined from the test split of the aforementioned action recognition datasets. It is important to emphasize that the expressions mined for testing are *completely* different from the ones used for cross-validation. Therefore, the proposed inference models have not seen any test frame or the same action pattern before, which provides an unbiased evaluation protocol. In order to provide robust estimators of performance, in the experiments of the current section, we repeat the data collection of 50 regular expressions and the test procedure steps ten times, reporting the mean and standard deviation of the evaluation metrics AUC

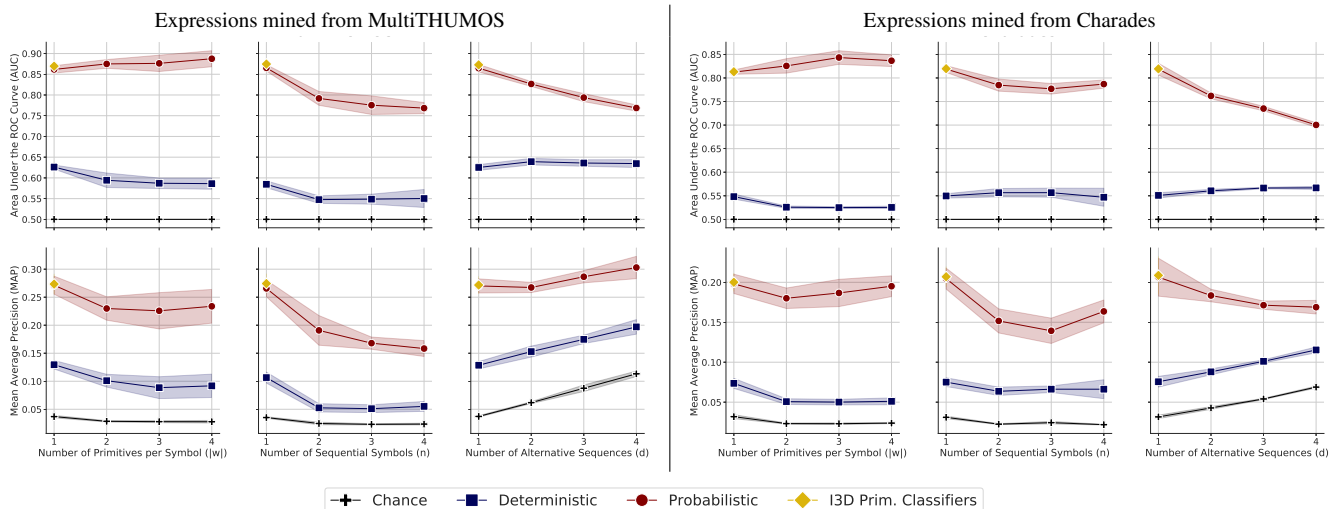


Figure 4. Comparison with standard action classification. Plots of the performance, in terms of AUC and MAP, of the proposed methods using the I3D model [1] as the primitive action classifier. We evaluate the models on collections of regular expressions of different complexity mined from the test videos of MultiTHUMOS and Charades datasets. These regular expressions follows the format defined in Equation 7 where all the variables are set to 1 with the exception of the one being evaluated. For instance, for the plot with variable number of sequential symbols (n) the expressions vary from (w_1^+) to $(w_1^+ \succ \cdots \succ w_4^+)$. Differently from the other experiments, the symbols here denote any subset that contains the primitives.

and MAP. Note that these metrics are computed over the recognition of the *whole complex activity* as a singleton label. They are *not* computed per primitive.

Comparison To Standard Action Recognition. Traditional action classification aims to recognize a single action in a video, making no distinction if the action is performed alone or in conjunction with other actions. Abusing the proposed regular expression notation, for now consider the symbols w in Equation 7 as the collection of all subsets of the primitive actions that contains the actions in w. For instance, the symbol $\{a_2, a_3\}$ represents (only here) the set of symbols $\big\{\{a_2, a_3\}, \{a_2, a_3, a_4\}, \ldots \{a_2, a_3, a_4, \ldots, a_{|\mathcal{A}|}\}\big\}$. Then, we can say that the traditional action classification problem is the simplest instance of our formulation where the input regular expressions are of the type $\{a\}^+$, meaning one or more frames depicting the action a alone or in conjunction with other actions. Therefore, starting from this simplified setup, we analyze how our models behave as we increase the difficulty of the problem by dealing with more complex regular expressions. More specifically, we start from this simplest form, where all the regular expression parameters are set to one, and evolve to more complex expressions by varying some of the parameters separately. Figure 4 presents the results on expressions mined from the Charades and MultiTHUMOS datasets where we vary the number of concurrent (columns 1 and 4), sequential (columns 2 and 5), and alternated actions (columns 3 and 6) by varying the number of primitives per symbol $|w|$, the number of sequential symbols n, and the number of alternative sequences d in the mined regular expression and video clip data, respectively.

Note that there is a significant difference in performance when compared to the results in Section 4.1. Such a difference is due to the quality of the primitive classifiers available for a challenging problem like action classification. For instance, the digits classifiers for the MNIST dataset are at least three times more accurate than the primitive action classifiers for Charades or MultiTHUMOS. However, different from the deterministic model, the probabilistic model is able to extend the primitive action classifiers, the I3D model, for complex expressions without degenerating the performance significantly. For instance, considering all setups, the probabilistic model presents a reduction in performance of at most 15% in both datasets and metrics used. This result suggests that the proposed model can easily scale up the developments in action recognition to this challenging compositional activity recognition problem.

Trimmed Compositional Activity Classification. We now evaluate the ability of the proposed algorithms to recognize specific activities in trimmed video clips which depict only the entire activities from the beginning to the end. Different from the previous experiment, but like the other experiments, the input regular expressions are formed by symbols that are only subsets of primitives. For instance, the symbol $\{a_2, a_3\}$ means that the primitive actions $a_2, a_3 \in \mathcal{A}$ happen exclusively in a frame. In addition, we mined test regular expressions with different combinations of parameters ranging jointly from 1 to 6. Table 1 presents the results.

We would like to emphasize the difficulty of the problem where the chance performance is only about 2% MAP in both datasets. The deterministic model works only slightly better than chance, which is also a consequence of the im-

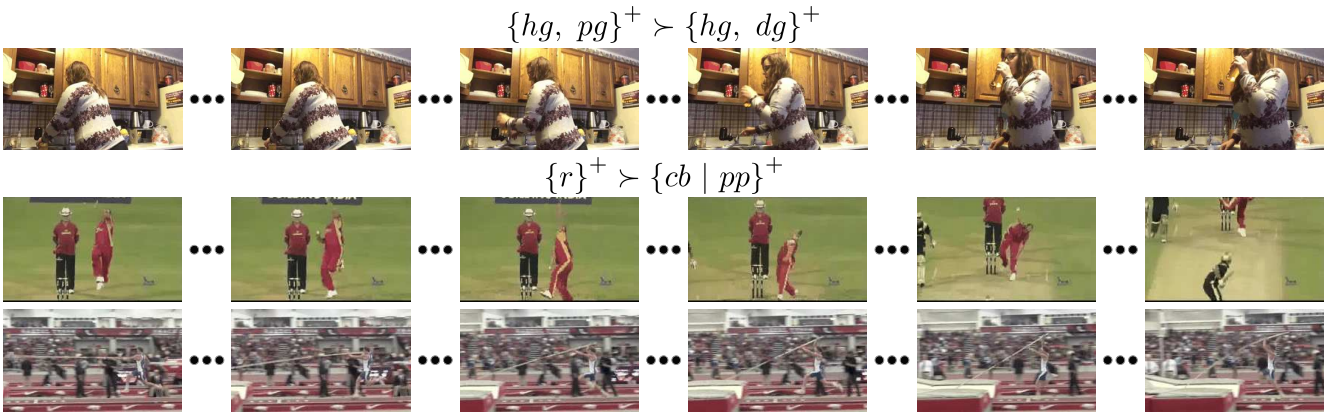


Figure 5. Examples of regular expressions and video clips matched with high probability by the proposed probabilistic inference model. The primitive actions used to form these regular expressions are holding a glass (hg), pouring water into the glass (pg), drinks from the glass (dg), running (r), cricket bowling (cb), and pole vault planting (pp).

perfect quality of the primitive classifiers due to the difficulty of action recognition as discussed before. On the other hand, the probabilistic model provides gains above 20% in AUC and 10% in MAP compared to the deterministic approach in both datasets. This shows the capability of the probabilistic formulation to surpass the primitive classifiers' imprecision even when the activity of interest is very specific, producing a very complex regular expression.

Table 1. Results for activity classification in trimmed videos.

	Expressions mined from MultiTHUMOS		Expressions mined from Charades	
Method	AUC	MAP	AUC	MAP
Chance	50.00 (±0.0)	2.00 (±0.00)	50.00 (±0.0)	2.00 (±0.00)
Deterministic	52.46 (±0.77)	3.66 (±0.48)	51.85 (±0.83)	4.40 (±1.15)
Probabilistic	73.84 (±2.63)	13.76 (±1.93)	74.73 (±2.35)	15.19 (±1.09)

Untrimmed Compositional Activity Classification. In this task, we evaluate the capability of the proposed models for recognizing specific activities in untrimmed videos which may depict the entire activity of interest at any part of the video. Here, videos can contain more than one activity, and typically large time periods are not related to any activity of interest. In this context, we modify the mined regular expressions to allow matches starting at any position in the input video. It is easily accomplished by doing the following transformation: $re \rightarrow .^\star re.^\star$ where $(.)$ is the "wildcard" in standard regular expression engines and in our formulation consists of every subset of primitive actions. In addition, we do not trim the video clips, instead we compute matches between the mined regular expressions and the whole video aiming to find at least one occurrence of the pattern in the entire video. We present the results on Table 2 where we compute matches between regular expressions and the videos that have at least one positive video clip for the set of mined regular expressions.

In the same fashion as the previous experiments, the probabilistic model performs significantly better than the deterministic model. More specifically, the performance of the probabilistic model is at least 10% better than the deterministic model in this experiment on both metrics and datasets. Therefore, the proposed probabilistic model is able to analyze entire videos and generate their global classification as accurately as it does with trimmed video clips.

Table 2. Results for activity classification in untrimmed videos.

	Expressions mined from MultiTHUMOS		Expressions mined from Charades	
Method	AUC	MAP	AUC	MAP
Chance	50.00(±0.0)	4.21(±0.20)	50.00(±0.0)	2.58(±0.01)
Deterministic	65.69(±1.34)	12.59(±1.32)	55.76(±1.21)	6.77(±1.20)
Probabilistic	75.96(±1.49)	26.03(±1.45)	75.43(±1.35)	17.90(±1.25)

Qualitative Evaluation. In Figure 5, we display examples of regular expressions and matched video clips using the proposed probabilistic model. In the first row, we see examples of concurrent and sequential actions where the woman depicted is holding a glass (hg) and pouring water into the glass (pg) simultaneously, and then she drinks from the glass (dg) while holding the glass. In the last two rows, we see an example of alternated actions where the desired action pattern starts with running (r) and finishes with someone either bowling (cb) or pole vault planting (pp).

5. Conclusion

In this paper, we addressed the problem of recognizing complex compositional activities in videos. To this end, we describe activities unambiguously as regular expressions of simple primitive actions and derive deterministic and probabilistic frameworks to recognize instances of these regular expressions in videos. Through a variety of experiments using synthetic data, we showed that our probabilistic framework excels in this task even when using noisy primitive classifiers. In the action recognition context, the proposed model was able to extend state-of-the-art action classifiers to vastly more complex activities without additional data annotation effort or large performance degradation.

Acknowledgements: This research was supported by the Australian Research Council Centre of Excellence for Robotic Vision (CE140100016).

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017). 1, 6, 7

[2] Alaa El-Nouby, Ali Fatouh, Ahmed Ezz, Adham Gad, Ahmed Anwer, and Ehab Albadawy. *Smart Airport Surveillance System (Action Recognition, Unattended Object Detection, Tracking)*. PhD thesis, 07 2016). 1

[3] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. *ICCV*, 2017). 1, 2

[4] Kirill Gavrilyuk, Amir Ghodrati, Zhenyang Li, and Cees GM Snoek. Actor and action video segmentation from a sentence. In *CVPR*, 2018). 1, 2

[5] Amirhossein Habibian, Thomas Mensink, and Cees GM Snoek. Video2vec embeddings recognize events when examples are scarce. *PAMI*, 39(10):2089–2103, 2017). 2

[6] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017). 1, 2

[7] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21, 2017). 2

[8] John Hopcroft. An n log n algorithm for minimizing states in a finite automaton. *Theory of machines and computations*, pages 189–196, 1971). 3

[9] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *CVPR*, 2019). 1, 2

[10] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Pic: Permutation invariant convolution for recognizing long-range activities. *arXiv preprint arXiv:2003.08275*, 2020). 1, 2

[11] Nazlı İkizler and David A Forsyth. Searching for complex human activities with no visual examples. *IJCV*, 80(3):337–357, 2008). 2

[12] Lucian Ilie and Sheng Yu. Constructing nfas by optimal use of positions in regular expressions. In *Annual Symposium on Combinatorial Pattern Matching*, 2002). 3

[13] Mihir Jain, Jan C van Gemert, Thomas Mensink, and Cees GM Snoek. Objects2action: Classifying and localizing actions without any video example. In *ICCV*, 2015). 1, 2

[14] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs. *arXiv preprint arXiv:1912.06992*, 2019). 2

[15] Soo Min Kang and Richard P Wildes. Review of action recognition and detection methods. *arXiv preprint arXiv:1610.06906*, 2016). 2

[16] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *PAMI*, 36(3):453–465, 2014). 2

[17] Sian Lun Lau, Immanuel König, Klaus David, Baback Parandian, Christine Carius-Düssel, and Martin Schultz. Supporting patient monitoring using activity recognition with a smartphone. In *International symposium on wireless communication systems*. IEEE, 2010). 1

[18] Mark V Lawson. *Finite automata*. Chapman and Hall/CRC, 2003). 3

[19] Daniele Liciotti, Marco Contigiani, Emanuele Frontoni, Adriano Mancini, Primo Zingaretti, and Valerio Placidi. Shopper analytics: A customer activity recognition system using a distributed rgb-d camera network. In *International workshop on video analytics for audience measurement in retail and digital signage*, 2014). 1

[20] Bingbin Liu, Serena Yeung, Edward Chou, De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Temporal modular networks for retrieving complex compositional activities in videos. In *ECCV*, 2018). 1, 2

[21] Jingen Liu, Benjamin Kuipers, and Silvio Savarese. Recognizing human actions by attributes. In *CVPR*, 2011). 2

[22] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943). 3

[23] Pascal Mettes and Cees GM Snoek. Spatial-aware object embeddings for zero-shot localization and classification of actions. In *ICCV*, 2017). 1

[24] Ruslan Mitkov. *The Oxford Handbook of Computational Linguistics (Oxford Handbooks in Linguistics S.)*. Oxford University Press, Inc., New York, NY, USA, 2003). 3

[25] AJ Piergiovanni and Michael S Ryoo. Learning latent super-events to detect multiple activities in videos. In *CVPR*, 2018). 2

[26] Michael O Rabin. Probabilistic automata. *Information and control*, 6(3):230–245, 1963). 1, 4

[27] Michael O Rabin and Dana Scott. Finite automata and their decision problems. *IBM journal of research and development*, 3(2):114–125, 1959). 3

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

[29] Robert Sedgewick and Kevin Wayne. *Algorithms*. Addison-Wesley Professional, 2011). 3

[30] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016). 2, 6

[31] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *ICML*, 2015). 5

[32] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015). 1

[33] Nam N Vo and Aaron F Bobick. From stochastic grammar to bayes network: Probabilistic parsing of complex activity. In *CVPR*, pages 2641–2648, 2014). 2

[34] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018). 1, 2

[35] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *IJCV*, 2017). 2, 6