Recognition of Multiple Overlapping Activities Using Compositional CNN-LSTM Model

Tsuyoshi OKITA
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata,
Kitakyushu-shi
Fukuoka, 804-8550, JAPAN
tsuyoshi.okita@gmail.com

Sozo INOUE
Kyushu Institute of Technology
1-1 Sensui-cho, Tobata,
Kitakyushu-shi
Fukuoka, 804-8550, JAPAN
sozo@mns.kyutech.ac.jp

UbiComp/ISWD' 17 Adjunct, September 11-15, 2017, Maui, HI, USA.

ACM 978-1-4503-5190-4/17/09$15.00.
https://doi.org/10.1145/3123024.3123095

Abstract
This paper introduces a new task, a recognition of multiple overlapping activities in the context of activity recognition. We propose a compositional CNN+LSTM algorithm. The experimental results show on the artificial dataset that it improved the accuracy from 27% to 43%.

Author Keywords
Activity Recognition; Deep Learning

ACM Classification Keywords
H.1.2 [User/Machine Systems]: Human information processing

Introduction
Recently, DeepSense model [6] and Deep convolutional LSTM model [3] propose a deep learning architecture for activity recognition. By reducing the dimensionality of the sensor outputs in the form of representation using Convolutional Neural Networks (CNN), it classifies the signal pattern by Recurrent Neural Networks (RNN). This paper focuses on the problem that these architectures [6, 3] as well as most of other architectures (the semi-CRF model [5] and the Bayesian model using the importance sampling [1]) recognize one activity at a time. We propose an algorithm which recognizes multiple overlapping activities.

	different activities by one person	same activity by different persons
wearable sensor	drink coffee & open drawer	N.A.
	(α)	
environmental sensor	drink coffee & open drawer	drink coffee by A & drink coffee by B
	(β)	(γ)

Table 1: Figure shows the possibilities of multiple overlapping activities depending on the sensor type.

In the following sections, we first examine the possibility of multiple overlapping activities in activity recognition. Then, we talk about an assumption that the sensor outputs reflect the multiple overlapping activities. Based on this assumption, we build a compositional CNN+LSTM algorithm. We show our experiments and conclude.

Multiple Overlapping Activities

Table 1 shows the possible three multiple activities: (α), (β), and (γ). First, on the wearable sensor, if two different activities which are among our target activities occur, this can be the first category (Category (α)).

For example, the 'drink coffee' activity and the 'open drawer' activity can be simultaneously taken place by the single person. Note that the same activity invoked by persons A and B is non applicable since this setting of one wearable sensor connecting multiple persons is physically impossible. [1] Second, on the environmental sensor, if two different activities occur, this can be the second category (Category (β)). For example, if single person executes the 'drink coffee' activity and the 'open drawer' activity at the same time, the environmental sensor may capture these activities. Third, on the environmental sensor, if the single activity can be invoked by person A as well as by person B at the same time. the sensor may capture these activities as the quantity which is twice as much as those which is invoked by person A alone.

One remark is that we exclude the *composite activities*[2] [4] from multiple overlapping activities. Two activities A and B are called composite activities when one activity is a part of the other activity.

One important aspect is that most of the previous datasets do not include multiple overlapping activities. First, this is due to the fact that multiple simultaneous activities were actually taken place but only a single activity at a time is labeled. The annotation process tends to ignore such complex annotation. Second, this is due to the psychological reluctance of participants or the data collectors. Even though the multiple activities are theoretically possible[3], the reluctance attitude of the participants prevented these cases from taking place in reality.[4]

Sensor Output Additivity by Multiple Activities

We assume that the sensor output reflects the number of activities.[5] Without loss of generality, we assume the simplest situation with the two sensor set-ups where the overall sensor outputs can be calculated by the sum of the sensor outputs of the first one and the second one. We illustrate this with the imaginative sensor with only one dimension. We assume that the same situation holds for more than three persons.

Definition 1 (Sensor Output Additivity Assumption) *Suppose that the activity A by person 1 yields the sensor output* $f_1(x)(t_1 \leq x \leq t_T)$ *and the activity A by person 2 yields the sensor output* $f_2(x)(t_1 \leq x \leq t_T)$*. If we have activity A by person 1 alone in* $(t_1 \leq x \leq t_T)$*, the*

[1] If one person wears the sensor, this sensor often works for this person alone. However, the environmental sensor may detect the behavior of several persons at the same time.

[2] Roggen et al. show an example, (a) the 'cutting bread' activity consists of several small activities, such as (b) 'reach for bread' activity, (c) 'move to bread cutter' activity, and (d) 'operate bread cutter' activity [4].

[3] The other category is the case when the multiple activities are theoretically impossible. For example, if the door and the table are in a distant place, the single person cannot do two activities physically which relate to these two distant object at the same moment

[4] The OPPORTUNITY datasets allow the activity category (α) [4] although this kind of activity did not occur probably due to this reluctance.

[5] This assumption depends on the type of sensors and we assume that we pick up the sensors that confirm this assumption. This assumption contrasts the setting with multi-label setting.

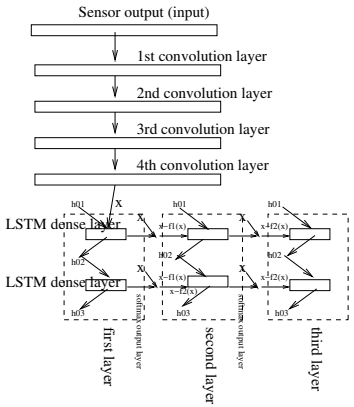

Figure 1: Figure shows the overview of compositional CNN+LSTM model. This model has three layers of LSTM which corresponds to the constraint that this model can capture at most three overlapping activities at a time. Theoretically, we can extend this number of layers in LSTM.

OPPORTUNITY(gesture)	
(Yao16)	89.9
(Ordonez16)	<u>91.5</u>
Ours	90.2
OPPORTUNITY(locomotion)	
(Yao16)	87.1
(Ordonez16)	<u>89.5</u>
Ours	88.7

Table 2: Performance on OPPORTUNITY datasets.

sensor output is $f_1(x)(t_1 \leq x \leq t_T)$. If we have activity A by person 1 and 2 together, we have the sensor output $(f_1 + f_2)(x)(t_1 \leq x \leq t_T)$.

One remark is about composite activities. In the previous section, we mentioned that we do not consider composite activities among multiple activities. Although it is possible that the activities A and B occur at the same time the sensor output additivity assumption does not hold since the sensor output does not reflect whether the activities A and B occur together or alone.

Compositional CNN+LSTM Model

We build an algorithm under the sensor output additivity assumption. When the feature quantity becomes more than the sensor output from one activity alone, we need to expect the possibility that multiple sensor outputs are combined (a compositional model [2]). Without loss of generality, we build a model which can recognize up to three multiple overlapping activities at a time.

Figure 1 shows an overview of our idea. The first four layers are convolution layers [3][6]. The feature maps $a_{ij}^{(i)}$ in i-th convolution layers are shown as in (1):

$$\text{i-th convolution layer(i-th feature map)} \quad \left\{ \; a_{ij}^{(i)} = \tanh_{(1)}((W^k, x_t)_{ij} + b_k) \right. \tag{1}$$

Our ideas are in LSTMs. Let us consider that the parameters of LSTMs are shared. The first layer of LSTM learns the interval which is an output of single event but the intervals which are outputs of multiple events as well. Both of these learn $f(x)$ of mixing status and output, and its inverse function $f^{-1}(y)$. Let C be the number of class. For the mixture of two components, we predict the class using a matrix $C \times C$ which is appeared in the training data.

First layer

$$\left\{ \begin{array}{l} h_{<t>}^{(1)} = \text{LSTM}_{(1)}(h_{<t-1>}^{(1)}, x_t; \tanh) \\ p(x_{t,j} = 1 | x_{t-1}, \ldots, x_1) = \frac{\exp(w_j^{(1)} h_{<t>}^{(1)})}{\sum_{j'=1}^{K} \exp(w_{j'}^{(1)} h_{<t>}^{(1)})} \end{array} \right.$$

Since this provides only the classification results and when the number of component is one, this shows $f(x)(= x)$. Hence, we learn at the same time $f(x|c) : h_i, x_i, c \rightarrow x$ as an internal auxiliary function depending on c.

i-th layer (i=2 and 3)

$$\left\{ \begin{array}{l} h_{<t>}^{(i)} = \text{LSTM}_{(i)}(h_{<t-1>}^{(i)}, x_t - f_{(i-1)}(x_t); \tanh) \\ p(x_{t,j} = 1 | x_{t-1}, \ldots, x_1) = \frac{\exp(w_j^{(i)} h_{<t>}^{(i)})}{\sum_{j'=1}^{K} \exp(w_{j'}^{(i)} h_{<t>}^{(i)})} \end{array} \right.$$

For the combinations which are not appeared in the training set, we prepare the example of output beforehand for the test phase. We let two components as $f_1(x)$ and $f_2(x)$ and we consider the sensor output additivity assumption, i.e. $f(x) = f_1(x) + f_2(x)$.[7] We prepare a tensor $C \times C \times C$ for the possible combination of three components for the second layer.

In a test phase, if the sensor output consists of single component or the mixture of already appeared components in the training example, we obtain the results of learning based on the training examples. If the sensor output consists of the mixture of components not appeared in the training examples, we obtain the results using the predicted results for the virtualized combination of components in the above.

[6] These four layers of convolution layers are the same as Ordonez et al.

[7] If two components have dependencies, the distribution of sensor outputs will become smaller than that of this for independent case. However, we omit this dependent case in this paper.

	A	B	C
none	763	745	769
single	210	228	205
multi	0	0	0
	D	E	F
none	0	0	0
single	893	865	878
multi	107	135	122

Table 3: Among 2 groups, the first group consists of data sets A, B, and C. Each number shows the counts of activities. *Single* denotes the single activity at a time while *multi* denotes the two activities at a time. *None* denotes no activities.

	A	B	C
baseline	68.2	66.1	70.5
compo	68.2	66.1	70.5
	D	E	F
baseline	14.0	14.6	14.4
compo	20.0	19.6	19.4

Table 4: Table shows accuracies. Baseline is CNN+LSTM while the compositional model is ours. In the experiments of D, E, and F, the compositional model performs better than the baseline model.

Experimental Results

Implementation is done by Theano / Lasagne. We have four layers of convolution layers, two layers of LSTM, and a softmax output layer. The size of the window is 24, the window step is 12, the final width of the convolution operation is 8, the number of filters is 64, the size of the filter is 5, the number of hidden layers of LSTM is 128, the batch size is 100, dropout is 0.3, and the number of epoch is 500. Experiments were performed on NVIDIA GTX 1080.

First, we prepare two sets of artificial data where each set has three kinds (Table 3). The first set has three kinds of data A, B, and C where the events occur either (1) one at a time (“single”) (30-40%) or (2) none at a time (“none”)(60-70%). The second set has three kinds D, E, and F where the events occur either (1) one at a time (“single”)(86%-89%) or (2) multiple at a time (“multi”)(11%-14%).

Experimental results are shown in Table 4. Since the data set of A, B, and C have single label which does not require the compositional instance model, the performance in the baseline and in the compositional instance model were identical. On the other hand, the data set of D, E, and F have multiple labels. The baseline system performs very bad compared with the first group. We could increase the performance using the compositional instance model from 27% to 43%. However, the averaged accuracy was around 20% which was not a satisfactory result.

Second, we show the performance on OPPORTUNITY datasets in Figure 2. Ours are slightly inferior to Ordonez et al.[8]

[8] Unfortunately, results show that this dataset does not capture the multiple overlapping activities although some overlapping activities such as ’drinking coffee’ activity with the ’open drawer’ activity are theoretically possible while most of the combinations are theoretically impossible.

Conclusion

In this paper, we introduce the compositional CNN+ LSTM model. As the experimental results show, our composition instance model increased the accuracy from 27% to 43%.

REFERENCES

1. Sozo Inoue, Naonori Ueda, Yasunobu Nohara, and Naoki Nakashima, "Mobile Activity Recognition for a Whole Day: Recognizing Real Nursing Activities with Big Dataset", UbiComp, pp.1269-1280, 2015.
2. James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, Zoubin Ghahramani, “Automatic Construction and Natural-Language Description of Nonparametric Regression Models”, AAAI, pp.1242-1250, 2014.
3. Francisco Javier Ordonez, Daniel Roggen. “Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. Sensors 16:115, 2016.
4. Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Gerhard Troster, Paul Lukowicz, Gerald Pirkl, David Bannach, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, and Jose del R. Millan. "Collecting complex activity data sets in highly rich networked sensor environments", INSS10, pp.233-240, 2010
5. La The Vinh, Sungyoung Lee, Hung Xuan Le, Hung Quoc Ngo, Hyoung Il Kim, Manhyung Han, and Young-Koo Lee, "Semi-Markov conditional random fields for accelerometer-based activity recognition", Springer, pp.226-241, 2011.
6. Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, Tarek Abdelzaher. “DeepSense: A Unified Deep Learning Framework for Time-Series Mobile Sensing Data Processing”. arXiv:1611.01942v1, 2016.