

**LETTER**

**TS-ICNN: Time Sequence-Based Interval Convolutional Neural Networks for Human Action Detection and Recognition**

Zhendong ZHUANG†, Nonmember and Yang XUE†a, Member

**SUMMARY** The research on inertial sensor based human action detection and recognition (HADR) is a new area in machine learning. We propose a novel time sequence based interval convolutional neural networks framework for HADR by combining interesting interval proposals generator and interval-based classifier. Experiments demonstrate the good performance of our method.

**key words:** human action detection and recognition, inertial sensor, interval proposals, interval-based classification, convolutional neural network

1. Introduction

Because of the popular of smart phones and wearable devices, inertial sensor based human action recognition (HAR) gradually plays an important role in intelligent health-care, smart environments, and homeland security [1], [2]. Most of researches recognize activities which are cyclical, simple and long-term, such as walking, running, stairs-down and so on. However, in reality, human activities are complex. It is desirable to detect and recognize some specific actions from a series of complex activities. For example, we want to detect and recognize the action of smash in a badminton game.

To our knowledge, there are few researches on inertial sensor based human action detection and recognition (HADR) [3]. Most of previous sequence recognition algorithms are based on fixed sliding window [4], [5]. However, in the detection of human actions, the intervals of different actions are not constant. So the fixed size window cannot precisely detect different intervals. To address this problem, in this paper, we propose a new efficient method which can accurately find out the valid intervals of different actions for time sequence based HADR.

In recent years, convolutional neural networks (CNNs) have improved great recognition performance not only for inertial sensor signal [6], [7], but for other time sequence, such as ECG signal [8]. CNNs can well capture the two most critical features in time sequence, local dependency and interrelationship. Furthermore, recent advances in image-based object detection are also driven by the success of CNNs, especially Faster R-CNN [9]. The clever use of CNNs in Faster R-CNN enlightens us and tells us how to use CNNs to implement region proposals and region-based classification.

Considering the characteristics of time sequence and the outstanding “region” idea in Faster R-CNN, we introduce an “interval” idea for time sequence. In this paper, a novel time sequence based interval convolutional neural networks (TS-ICNN) framework is proposed, as shown in the low half of Fig. 1. Our framework is composed of two sub-networks, interval proposal network and interval-based classification network. The first part (interval proposal network, IPN) generates precise and high-quality interval proposals over the whole sequence stream to tell the last part where to look. The last part (interval-based CNN, ICNN) identifies the actions based on these proposals.

2. Our HADR Style

For our HADR, the activities are tracked with a smart watch embedded with inertial sensor and recorded by sensor signals. These signals are a sequence of discrete values sampled at a certain sampling frequency accompanied with stochastic noise, as shown in the upper half of Fig. 1 (a). In contrast to HAR task, there are some challenges for HADR to get a good result. First of all, the sensor data are gathered only from wrist instead of various body parts as before [10], [11]. This measure is very close to human regular life, but doubtlessly provides less valuable information for feature extraction. In addition, the sensor signals of the target actions are very confused with those of other actions. For instance, the signals of “smash” are easily recognized as “swing” in badminton games. Last but not least, to our best knowledge, there are relatively few public datasets for benchmarking. Therefore, we collected our own dataset of badminton games for training and testing.

In order to collect dataset, 12 participants (8 males and 4 females) were asked to wear the smart watch on their right wrists and play badminton around 5 minutes. Meanwhile, we recorded a video for each one for later labeling. Swing and delivery are selected from a badminton game as our target actions, which occur frequently.

3. TS-ICNN for Inertial Sensor Based HADR

In this paper, we present TS-ICNN for HADR, which implements interval proposals and interval-based classification as IPN and ICNN sequentially. The bottom of Fig. 1 illustrates the overall pipeline of this network. Instead of using...
fixed sliding window, each sub-network takes entire time sequence as input and uses a convolutional (conv) network to extract features. By utilizing them, IPN predicts confidence (bg, fg) and bounding interval regression (ctr, lh) for each anchor to generate proposals for ICNN. ICNN captures corresponding features of interval proposals from global features and pools them into a fixed size by interval of interest (IoI) pooling layer. Finally, processed related features are followed by fully connected layer and soft-max layer for classification.

3.1 Data Pre-Processing

When an entire sequence is used as input of TS-ICNN for training, the trained model will over-fit since a sequence contains too much target actions. Additionally, this operation needs a large computing resources. To balance it, we regarded 10s as a process unit and divided each sequence into 10s segments during training. Moreover, we normalized these sequence segments respectively and utilized Gaussian filter to smooth them and reduce the noise. Finally, we padded two rows of fixed values between acceleration and angular velocity to separate them to prevent disturbing when applying conv kernel to extract feature [12].

3.2 Convolutional Network

With above processing, the input of conv network is an 8x500 matrix. For better performance, rectangular conv kernel is applied in building conv network. Figure 2 illustrates its structure. In this conv network, horizontal rectangular conv kernel (like Conv1, kernel size is 3x7) can enlarge the receptive field in time domain without the limit of row number. Simultaneously, replacing pooling layer in vertical with vertical rectangular conv kernel (like Conv2, kernel size is 4x1) helps to enhance the ability of capturing more complex relations between various sequences. These are key bottlenecks in features extraction of sequence data. Although IPN and ICNN have their own conv network, the output of their conv networks both are feature series, the feature maps with specific size like (1,x,y) (in Fig. 2). Furthermore, their first several conv layers are designed alike (in Fig. 2), which can be shared to reduce computing cost.

3.3 IPN and Anchor

Inspired by region proposal network of Faster R-CNN, we designed our own interval proposal network (IPN) with CNN, special for time sequence data. Note that, IPN uses large width kernel in several conv layers. Hence, each point of feature series has a large receptive field on the in-
putted time sequence segment with high-level semantics. As shown in Fig. 3, IPN predicts several interval proposals at each point of feature series, and denotes by $k$ the number of possible proposals for each point. So the $cls$ layer outputs $2k$ scores for each point to infer whether each anchor is positive (intersection-over-union overlap higher than 0.5 with any ground truth interval) or not, and the $reg$ layer has $2k$ outputs refining the coordinates of $k$ intervals. Especially, to product above scores, we used two sibling 1x1 convolutional layers with $2k$ channels to process feature series.

An anchor is centered at the center of relative receptive field and corresponding to a scale (in Fig. 3). For each anchor, it gets confidence ($bg, fg$) from $cls$ layer and bounding interval regression ($ctr, lh$) from $reg$ layer. $fg$ represent the probability that anchor is positive and $bg$ is the opposite. ($ctr, lh$) is computed to final proposal with formula (1), represented by the star point and end point $(x_1^*, x_2^*)$.

\[
\begin{align*}
  x_1^* &= 0.5l + x_1 + ctr \times l - 0.5e^{lh}l \\
  x_2^* &= 0.5l + x_1 + ctr \times l + 0.5e^{lh}l
\end{align*}
\]  

$(x_1, x_2)$ are the star point and end point of anchor, and $l$ ($l = x_2 - x_1$) is the length of anchor.

3.4 The IoI Pooling Layer

As an adaptive pooling layer, the IoI pooling layer captures the feature segments inside any interval proposals from $conv$ feature series and converts them into a small feature series with a fixed length $L$ by max-pooling, where $L$ is layer parameter. In this paper, each feature segment interval is expressed as a two-tuple $(x_1^*, x_2^*)$, the results of proposal interval $(x_1, x_2)$ dividing the ratio of the length of framework input to $conv$ feature series.

With computing the length $l'$ of feature segment ($l' = x_2^* - x_1^*$), the IoI pooling layer splits the feature segment into $L$ cells of smaller segment of approximate length $\frac{l'}{L}$, and then outputs the maximum over each small segment into the associated output cell. In addition, this special pooling operation is applied independently to each feature series channel.

3.5 Loss

With above definitions, our loss function is designed as:

\[ L_{all} = L_{IPN} + L_{ICNN} \]  

$L_{IPN}$ is the loss function of IPN, which consists of two parts. The former is the loss of classification; the latter is the loss of interval regression. $L_{ICNN}$ is soft-max cross entropy loss.

\[ L_{IPN} = \frac{1}{N} \sum_{i} N_{re} L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_{i} N_{p} L_{reg}(r_i, r_i^*) \]  

Here, $i$ is the index of an anchor (the total number is $N$), $p_i$ is its confidence $fg$ and $p_i^*$ is the ground truth. For the classification loss, $L_{cls}$ is soft-max cross entropy loss over two classes (object vs. not object). Thus, $p_i^*$ is 1 when anchor is positive and 0 when anchor is negative. In addition, the setting of $p_i^* L_{reg}$ can let regression loss be disable when anchor is negative. On the other hand, when the anchor is positive (the total number is $N_{reg}$), $L_{reg}$ is defined as

\[ L_{reg} = R(r_i - r_i^*) \]  

and $R$ is the robust loss function (smooth $L1$) defined in [13]. In (3), $r_i$ and $r_i^*$ are both vector. $r_i$ is $(ctr, lh)$, and $r_i^*$ is $(ctr^*, lh^*)$, which are computed by (4).  

\[
\begin{align*}
  ctr^* &= \frac{0.5l' + x_1^* - (0.5l + x_1)}{l} \\
  lh^* &= \log(l' / l)
\end{align*}
\]  

$(x_1^*, x_2^*)$ are the star point and the end point of near ground truth interval, $l'$ is its length.

4. Experiment

To evaluate the performance of TS-ICNN, we constructed experiments with Faster R-CNN on our dataset. We divided the dataset into 4 folds (each fold contains the data of 2 males and 1 female respectively) and successively used one fold for testing, and the rest folds for training. Considering our framework contains two sub-networks, we used a step-by-step method to train it. At the first step, we trained IPN until it had a stable loss. Then we trained IPN and ICNN together at the next step. Especially, the inputted proposals for ICNN are not only generated by IPN during training, but also slightly shifting or resizing ground truth. For testing, we evaluated detection mean average precision (MAP) over testing set as it is the common metric for detection.

The results of TS-ICNN and Faster R-CNN: Faster R-CNN is used for image-based object detection. In order to use it for time sequence of HADR, we utilized Faster R-CNN to handle the visualized sensor data, and its $conv$ network is ZF net [14]. We plotted each sensor data segment as a fixed size multi-series spline, as shown in Fig. 1 (a). Meanwhile, we ignored the normalization of sensor data as we aimed to use the difference of amplitudes among variates as feature, which is called “local outline information”. Table 1 shows that the proposed framework achieves an accuracy of 86.98%, which means TS-ICNN is feasible. Somewhat surprising, the accuracy of Faster R-CNN is 69.93%. Therefore, using CV method to handle visualized sequence may also work great.

Experiments on IPN: In this section, we first evaluated
the quality of interval proposals, which generated by IPN. To do this, we took top-ranked 5 proposals in the list of confidences $f g$ with non-maximum suppression (NMS) as IPN output, and counted their recall precision for ground truth. The results 94.87% shows that the IPN do a good job in proposals generation. Next, we set contrast experiments to investigate the sharing of convolution layer as a useful measure. This requires our IPN and ICNN share their conv layer. For training, we multiplied $L_{IPN}$ a balancing constant in the second step. The MAP drops to 82.75%, as shown in Table 1. This suggests that the training of IPN slightly influences conv layers to extract feature for ICNN, however, it reduces the cost of IPN a lot.

**Experiments on ICNN:** To better investigate ICNN’s effects on the interval-based classification alone, we replaced the original input of ICNN generated by IPN with real target actions’ interval during testing. And we set up the second experiment that used the real intervals as R-CNN’s input, too. Especially, since the quality of proposals of Faster R-CNN is worse than TS-ICNN, we fine-tuned R-CNN with new input. The results of these two experiments are 90.55% and 83.11% respectively. If an interval is perfectly extracted, ICNN is superior to R-CNN and has a good result.

**Vertical conv layer vs max-pooling layer:** We had demonstrated that TS-ICNN benefits more from replacing max-pooling with vertical rectangular conv kernel. This observation was investigated as follow. With only replacing vertical conv layer with max-pooling layer, we used the same training strategy. Table 1 shows the results. The max-pooling decreased the MAP from 86.98% to 70.44%, suggesting the vertical rectangular conv kernel actually enhance the ability of conv network to capture more high-level relation between different sequences.

### 5. Conclusion

Inertial sensor based human actions detection and recognition (HADR) from a series of complex activities is a new field of pattern recognition. In this paper, we present a novel TS-ICNN system to address the task of detection and recognition. In this framework, IPN shows considerable detection performance on our badminton games dataset. With high-quality detection, ICNN does well in actions recognition, too. TS-ICNN delivers a good result on HADR, which indicates that the combination of interval proposal method and interval-based classification method has a great power on detection and recognition of short-term actions among complex activities expressed in the form of time series. In addition, the vertical rectangular convolutional kernel enhances the ability of feature extraction between different sequences.

Although we only used TS-ICNN to achieve HADR, we believe it can serve as a basic and bottom method for other time sequence detection and recognition. More potential applications require further investigations.

### Acknowledgments

This work was supported in part by National Natural Science Funds (no. 61771199), Guangdong Science and Technology Research Plan (no. 2015B010131004, 2017A010101027, 2016A010101014) and the Fundamental Research Funds for the Central Universities of China (no. D2171560).

### References

