Online dense activity detection

Li Weiqi | Wang Jianming | Liang Jiayu | Jin Guanghao | Chung Tae-Sun

1School of Computer Science and Technology, Tianjin Polytechnic University, Tianjin, China
2Tianjin Key Laboratory of Autonomous Intelligence Technology and System, Tianjin Polytechnic University, Tianjin, China
3Tianjin International Joint Research and Development Center of Autonomous Intelligence Technology and System, Tianjin Polytechnic University, Tianjin, China
4Department of Computer Engineering, Ajou University, Suwon, South Korea

Abstract
Dense activity detection is a subtask of activity detection that aims to localise and identify multiple human activities in video clips. Existing methods adopt offline frameworks that require video frames to be available when activity detection begins. These offline methods are unable to be applied to online scenarios. An online framework is proposed for dense activity detection. The framework has two stages: warm-up and detection. Warm-up is the initialisation of dense activity detection, which generates a contextual model called an online aggregated-event. After that, the method moves into the detection stage, which consists of two modules: coarse label prediction and refined label prediction. Coarse label prediction predicts activity labels by taking the online aggregated-event as a prior; then, prediction is refined by two techniques, human–object interaction detection and online relation reasoning. The proposed method is evaluated using two dense activity datasets: Charades and AVA. The experimental results show that the proposed method has better performance than existing offline methods after the whole video input is added to the algorithm.

1 | INTRODUCTION

Human activity detection is the task of localizing and identifying human activity in untrimmed videos. It has several applications in fields such as robot visual surveillance, human–computer interaction, and intelligent robot navigation. Dense human activity detection is a subtask of human activity detection. The word ‘dense’ in this instance means that several human activities overlap each other over a certain time duration; thus, the activities are dense at that particular time in this video. In Figure 1(a), playing basketball can be broken down into several sequential activities. Because the activities are scattered, they are not dense; therefore, we call this single activity detection. Single activity detection produces only a single activity label for a certain video frame. An example of dense activity is shown in Figure 1(b), in which the man is sitting on a sofa while he is watching a laptop and eating some food. In the literature, a limited number of works are related to dense activity detection [1–4]. Unlike single activity detection, dense activity detection can generate multiple labels for a video clip and supply more details about the video being analysed. Dense activity detection was first proposed by Yeung et al. [2], and the authors claimed that a comprehensive understanding of human activity in videos requires multiple labels to be densely placed in a video sequence. To evaluate their methods, the authors extended the existing THUMOS dataset and built a new dataset named MultiTHUMOS. Nauata Junior et al. [4] suggested that the visual content in a video could be assigned fine-grained labels describing major components and coarse-grained labels depicting high-level abstractions; they used bidirectional inference neural networks and structured inference neural networks to perform structured label inference for visual understanding. By designing temporal structure filters, Piergiovanni et al. [1] introduced the concept of learning latent superevents in activity videos and used it for multiple activity detection. A superevent is a set of multiple activities occurring together in videos with a particular temporal pattern. It provides a temporal context for the detection of each single activity. Dai et al. [3] proposed a hierarchical structure called temporal aggregation network (TAN) for dense human activity.
recognition. By stacking spatial and temporal convolutions repeatedly, TAN forms a deep hierarchical representation to capture spatiotemporal information in videos.

The techniques of video analysis can be separated into two types based on their application scenarios: online and offline methods. As far as we know, all dense activity detection methods adopt offline frameworks.

To the best of our knowledge, this work is the first to propose an online framework of dense activity detection. Compared with offline methods, the implementation of online methods is more challenging, and dense activity detection is more challenging than single activity detection because:

1. All video frames have been available to offline methods since the beginning of detection, so dense activity detection can be formulated as a classification problem. For online methods, video frames are input in temporal order, and only the partial video can be observed during the process of activity detection. Even more difficult, online methods need to find the start of an activity as soon as it shows up in a video.
2. Dense activities are similar to each other because they usually occur in the same background. For an example, watching on a laptop, eating food and sitting on sofa take place in a living room, and their features of video frames have smaller between-class variability compare with playing basketball, which has a different background. Therefore, this makes the online label prediction of dense activities more difficult.

We propose an online dense activity detection method. The online aspect is defined as being able to detect activities from partially observed videos; the dense part is for videos consisting of more than one activity, and more than one activity label is assigned to one video frame. Included videos are for a single person performing multiple activities. Altogether, this presents a challenging task for a computer vision system to master. To tackle these difficulties, we divide the frameworks into two stages: warm-up and detection.

The warm-up stage is the initialisation of the algorithm. At the beginning of detection, too few frames are available for making the correct label prediction. The proposed method does not output activity label in the warm-up stage, but it builds a context model named the online aggregated event (OAE).

The detection stage consists of two modules: coarse label prediction and refined label prediction. First, the method creates label prediction by using OAE a priori in coarse label prediction. To improve the accuracy of label prediction, two techniques are adopted in refined label prediction. The technique of human–object interaction (HOI) is used to refine label prediction by considering the interaction between human and objects. The other technique is called online relation reasoning (ORR); it can refine label prediction by applying the temporal constraint of human activity across video frames. Our contributions are thus that:

- We are the first to propose a two-stage online framework of dense activity detection that can be applied to real-time applications.
- We introduce a new concept, the OAE. The OAE is generated by accumulating past dense activities. It forms a temporal context and helps to perform activity label prediction as a priori condition.

The rest of the work is organised as follows. In Section 2, we review the relevant literature; Sections 3 and 4 cover the technical details of our approach. We report results in Section 5 and conclude with suggestions in Section 6.

2 | RELATED WORK

Human activity detection is a popular research topic in computer vision. Several documented handcrafted, feature-based methods have given excellent results on many benchmark datasets. There has been increasing work on learning features for activity detection using convolutional models.

Based on different grouping criteria, human detection methods can be divided in single or dense activity detection methods and online or offline activity detection methods. In this section, we briefly review these types of methods.

2.1 | Single activity detection

Single activity detection produces only one label for one video clip; the methods of both offline frameworks [5, 6] and online frameworks [7–10] have been documented.

Offline single activity detection: Most existing single activity detection models follow the offline approach. They classify and localise activities after completely observing an entire video sequence [5, 11–13]. These methods treat detection as a classification problem. An instance of activity is classified after the entire temporal segment has been localised. They generally follow a two-stage pipeline and then add different postprocessing steps. The two-stage pipeline first segments the entire video into...
possible clips termed activity proposals, whereas the excluded temporal part is considered a background fragment that does not contain activity. Then, specific activity classifiers are applied to proposals to get segment-level predictions. Zhao et al. [5] proposed a structured segment network based on a two-stage pipeline. They divided complete activity into three phases: starting, course, and ending. Then built a structured classifier for activity completeness judgement to filter incomplete proposals. Gao et al. [11,12] revised the temporal boundary of the proposals by regression. In addition, Dai et al. [13] proposed a representation that captures context around a proposal and then ranked proposals to get high-quality proposals. Xu et al. [14] constructed an end-to-end three-dimensional (3D) network called R-C3D to process the video input. The R-C3D could also be called the video version of Faster convolutional neural networks (R-CNN). Other methods predict activities at the frame level, which aggregate frames of the same class to represent an instance of activity in video. Shou et al. [6] proposed a convolutional-de-convolutional network, which added a de-convolution layer after the C3D network to obtain frame-level detection results.

**Online single activity detection:** Online activity detection aims to predict activity categories, the start of activity, and temporal boundaries from partially observed videos. Unlike offline detection, in which current, previous and future information from a video is available, online detection lacks future information. Online activity detection was first proposed in De Geest et al. [7], who introduced a realistic dataset for online detection and an evaluation protocol to assist their research. Most methods [7–9] used the combination of CNNs and recurrent neural networks (RNNs) to implement online detection; they extracted the features of each frame with a CNN and used RNN to load historical information during video progress. The output of the RNN reflects the activity status of the current frame. Soomro et al. [10] proposed a person-centric and online approach to the challenging problem of localization and the prediction of activities and interactions in videos, which could localize activities and interactions in both the temporal and spatial domains. However, existing online methods can detect only a single activity instance in a segment.

### 2.2 Dense activity detection

The concept of dense activity detection was first proposed by Serena et al. in 2018 [2]; the authors argued that single activity detection had several limitations. First, a single description is often insufficient to describe the activities of a person fully. Second, single activity detection methods largely ignore the intuition that activities are intricately connected.

Yeung et al. [2] extended the THUMOS sports video dataset to build a new dataset, MultiTHUMOS, containing dense, multiple labels for each frame. In addition, a network based on long short-term memory (LSTM) was proposed to capture temporal relations within and between activity classes, which is useful in multiple activity recognition. The proposed MultiLSTM method outperformed the much simpler LSTM network on the MultiTHUMOS dataset in multiple activity detection. Piergiovanni et al. [1] introduced a concept called a superevent, which refers to a set of activities occurring together in one video. Specifically, they designed temporal structure filters to capture certain video intervals, which are then processed by a soft attention mechanism to learn supervenient representations. The supervenient representations are combined with per-frame features extracted by a CNN to provide frame-level representations and input to an activity detector to get the frame-level annotations. Dai et al. [3] proposed a network architecture, TAN, which combines the advantages of previous work [15,16]; it decomposes 3D convolutions into spatiotemporal aggregation blocks and aggregates multiscale spatiotemporal information without dramatically increasing network complexity. Carreira and Zisserman [17] introduced a two-stream inflated 3D ConvNet (I3D) which extends the 2D CNN Inception-V1 into 3D convolutions and helps aggregate multiscale spatiotemporal information for activity detection, especially in videos containing multiple activities. In Sigurdsson et al. [18], the method proposed by that author [19] is implemented as the baseline in the dense activity detection task. It uses the VGG-16 model architecture for both networks and follow the two-stream strategy. Ghosh et al. [20] proposed stacked spatiotemporal graph convolutional networks (Stacked-STGCN) for localising a sequence of activities over long videos. The stacked-STGCN leverages the spatial connection between the joints of the human body and connects those joints across time to form a spatiotemporal graph, which helps to improve the accuracy of activity localization.

Compared with single activity detection, dense activity detection has obvious advantages when it comes to understanding videos. However, there is still little documented research concerning dense activity detection [1,2]. We have also found no documented research regarding online dense activity detection.

## 3 | METHODOLOGY

In offline frameworks, the input is a complete video \( \{x_i\}_{i=1}^{L} \). In online dense activity detection, the input video is observed partially.

The pipeline of this framework is shown in Figure 2.

**Input:**

The input of the framework is denoted as \( \{x_i\}_{i=1}^{L} \), where \( x_i \) is the \( i \)-th frame in the video sequence; \( x_i \) is the most recently updated frame. The input frames \( \{x_i\}_{i=1}^{L} \) are grouped into sliding windows, \( W_1, W_2, \ldots, W_t \) (\( t \) represents the order of sliding windows) and the sliding windows do not overlap each other.

**Output:**

The output of the framework is denoted as \( \overline{\{\overline{y_i}\}}_{i=1}^{t}, \overline{y_i} = \{\overline{y_i^1}, \overline{y_i^2}, \ldots, \overline{y_i^C}\} \), and \( \overline{y_i^j} \) is a score vector of \( C \)
where predefined activity classes, which is calculated by the algorithm for \( i \)-th frame \( x_i \). If \( \vec{y}_i = \max\{\vec{y}^c\} \), activity in \( x_i \) is recognized as the \( c \)-th predefined class.

### 3.1 Warm-up stage

In the warm-up stage, the proposed method mainly computes the OAE model representation. First, an online window sampling strategy (detailed in Section 4.1) is proposed to generate the aggregated window \( W^{af} \). We notate \( W^{af} \) as \( \{x_{ij}\}_{j=1}^{N} \). The output of OAE is notated by \( E_i \), which is a matrix of the \( c \)-th pre-defined class (more details are provided in Section 4.1):

\[
E_i = f(\{x_{ij}\}_{j=1}^{N}), \quad c = 1, 2, \ldots, C \quad i = \lfloor t/N \rfloor \quad (1)
\]

### 3.2 Detection stage

#### 3.2.1 Coarse label prediction

In the coarse label prediction module (Section 4.2), we perform per-frame feature extraction on each frame of input \( x_i \). By taking the OAE \( E_i \) as the context a priori, per-frame label prediction is made by:

\[
\vec{y}_i^1 = f_1(E_i, x_i), \quad i = t_0, \ldots, t; \quad t > t_0 \quad (2)
\]

where \( t_0 \) represents the time when the detection stage begins.

#### 3.2.2 Refined label prediction

Two techniques are adopted in the refined label prediction module. HOI is used to refine label prediction by considering the interaction between humans and objects in the current video frame; ORR refines label prediction by applying the temporal constraint of human activity across video frames.

**HOI module** (Section 4.3): Some human activities are related only to the human body and its pose. However, other human activities involve objects, and the relation between humans and objects in such instances includes informative cues for recognising human activities. The HOI module is adopted to improve the performance of activity detection by considering the interaction between humans and objects. The HOI module is illustrated in Equation (3):

\[
\vec{y}_i^2 = f_2(\vec{y}_i^1, x_i), \quad i = t_0, \ldots, t; \quad t > t_0 \quad (3)
\]

**ORR module** (Section 4.4): The ORR module is used to solve the problem of different activities that may have different temporal durations. It can also help to remove noise from frame-level predictions.

After that, we get the frame-level activity prediction results by:

\[
\vec{y}_i^2 = f_3(\{x_i\}_{i=t_0}^{t}, \{\vec{y}_i^2\}_{i=t_0}^{t}), \quad i = t_0, \ldots, t; \quad t > t_0 \quad (4)
\]

### 4 ALGORITHM

#### 4.1 Online aggregated-event model

OAE is an online variant of latent superevent representation introduced in the work [1]. The main purpose of OAE is to summarise historical frames and generate a context model, which is taken as a priori for frame-level label prediction. The output of OAE is calculated in the warm-up stage; it also will be updated repeatedly in the detection stage. To summarise the
received frames, we design a window sampling strategy to generate the input to the OAE model. Here, the input is called the aggregated window.

The window sampling strategy is shown in Figure 3. We make an aggregated window of identical length with the sliding windows.

1. We initialise the aggregated window by \( \mathbf{W}_1 = \mathbf{W}_1 = \{ x_i \}_{i=1}^N \).
2. We update the aggregated window \( \mathbf{W}_{i+1} \) with \( \mathbf{W}_{i+1} \) and \( \mathbf{W}_1 \); the frames of the first half aggregated window \( \mathbf{W}_{i+1}^{even} \) are the even frames of the previous aggregated window \( \mathbf{W}_i \); the frames of the second half aggregated window \( \mathbf{W}_{i+1}^{od} \) are the even frames of the sliding window \( \mathbf{W}_{i+1} \).

The length of aggregated windows could be greater than the length of sliding windows \( N \) to improve summarising historical frames, although we make it identical to \( N \) here.

The input to OAE is \( \mathbf{W}_i = \{ x_i \}_{i=1}^N \), and the output is the aggregated-event matrix \( \mathbf{E}_i \). The elementary form of the aggregated-event matrix \( \mathbf{E}_i \) is described in Equation (5):

\[
\mathbf{E}_i = \sum_{j=1}^{N} f_c \cdot \phi_{3D}(x_{ij})
\]

where \( \phi_{3D}(x_{ij}) \) is a 3D CNN feature descriptor called I3D, which has been pre-trained on ImageNet [17]. \( f_c \) is the filter coefficient of the predefined \( c \)-th class, which is calculated by temporal structure filters [1]. The temporal structure filter is used to capture the temporal context using historical frames. It is an extension of the spatial attention model proposed in Gregor et al. [21]. The previous attention model repeats a single Gaussian distribution several times with a fixed stride, whereas a temporal structure filter learns several independent distributions [1].

Because dense activities can share the same aggregated event, it makes sense to learn a set of \( M \) different temporal structure filters and share these filters across the classes. \( M \) is less than the number of classes \( C \). To represent the aggregated event of each activity class \( c \) using such \( M \) filters, we learn a set of per-class soft-attention weights, allowing each activity class to select some of the \( M \) structure filters for use. For a set of \( C \) classes, we learn weights \( \omega_{c,m} \) and compute the soft-attention as Equation (6):

\[
A_{c,m} = \frac{\exp(\omega_{c,m})}{\sum_{c} \exp(\omega_{c,c})}
\]

We then calculate the aggregated-event matrix \( \mathbf{E}_i \) by applying these weights to the \( M \) temporal structure filters:

\[
\mathbf{E}_i = \sum_{m} A_{c,m} \cdot \sum_{j=1}^{N} f_m[j] \cdot \phi_{3D}(x_{ij})
\]

Details of the OAE calculation are shown in Algorithm 1.

![Figure 3](image-url) Window sampling strategy \((N = 8)\). In the figure, sliding windows are orange and aggregated windows are blue. The aggregated window \( \mathbf{W}_i \) is built from \( \mathbf{W}_3 \) and \( \mathbf{W}_4 \). The frames of the first half aggregated window \( \mathbf{W}_{i+1}^{even} \) are the even frames in \( \mathbf{W}_i \), which are \( x_{i0}, x_{i2}, x_{i3}, \) and \( x_{i5} \); the frames of the second half aggregated window \( \mathbf{W}_{i+1}^{odd} \) are the even frames in \( \mathbf{W}_i \), which are \( x_{i1}, x_{i3}, x_{i4}, \) and \( x_{i6} \).

### 4.2 | Coarse label prediction module

In this module, the learnt OAE matrix \( \mathbf{E}_i \) is used as the context a priori for conducting frame-level activity detection on \( x_t \). Specifically, the per-frame binary classification is conducted for each activity by incorporating the OAE with the per-frame CNN representation \( \phi_{3D}(x_t) \):

\[
\overrightarrow{y}_t^i = f_1(E_t^i, x_t),
\]

\[
f_1 = \sigma(H[E_t^i, \phi_{3D}(x_t)]),
\]

\[
i = t_0, \ldots, t_t; \quad t > t_0, \lfloor t/N \rfloor
\]

where \( E_t^i \) is the OAE of activity \( c \), \( H \) is a learnable parameter, and \( \sigma \) is the sigmoid function; \( \phi_{3D}(x_t) \) refers to the features of the \( i \)-th frame. Details are shown in Algorithm 2.

### 4.3 | Human–object interaction detection module

In this section, spatial information is used to refine the results. We take single frame \( x_t \) and the results of the coarse label prediction module \( \overrightarrow{y}_t^i \) as input. The ambiguity of activity prediction is eliminated by examining the interaction and relation between instances in the image.

The HOI detection model is a method in Gao et al. [22] to deal with the relations among different actions during prediction. It is based on the object detection results of Faster R-CNN [23] and been validated on HICO-DET dataset whics the subset of [24]. The model dynamically generates an attention map conditioned on object–human instances of interest, which highlights relevant regions in an image that may be helpful for recognising HOI associated with given instances. For example, the human-centric attention map often focuses on surrounding objects; this helps, considering that multiple interacting objects also enrich the detection of dense activities. On the other hand, because different interactions with same objects often occur, such as ‘hold a cell phone and looking at it’ and ‘hold a cell phone
without looking at it’, the object-centric attention map highlights informative features from the human body and helps to disambiguate action prediction for the detected person.

Single frame \(x_i\) is input to the algorithm. First, we get instances and corresponding scores \([b_{ho}, s_{ho}], [b_{bo}, s_{bo}]\), in which instances are demarcated by detection box \(b_{ho}, b_{bo}\) for humans and objects, respectively.

Then, we predict activities based on the appearance of detected instances. The corresponding prediction result score is \(s_{ho}, s_{bo}\) for humans and objects, respectively.

For each detected human–object bounding box pair \((b_{ho}, b_{bo})\), we get the activity prediction score \(s_{att}\) based on the instance-centric attention network.

Finally, we combine the results based on their appearance with the results based on interaction to obtain activity prediction results \(\overrightarrow{y}_{i}'\):

\[
\overrightarrow{y}_{i}' = s_{ho} \cdot s_{bo} \cdot (s_{ho} + s_{bo}) \cdot s_{att}
\]  

A more detailed inference is documented in Gao et al. [22]. Then, the activity prediction results are refined by average coarse label prediction scores \(\overrightarrow{y}_{i}^{l}\) and HOI prediction scores \(\overrightarrow{y}_{i}^{m}\). Details are shown in Algorithm 3.

4.4 | Online relation reasoning module

In this section, we refine frame-level label prediction by ORR, an online variant of temporal relational network (TRN) [25]. TRN is an effective and interpretable network module that can consider temporal relations between video frames at different time scales. Zhou et al. [25] claimed that TRN-equipped networks can accurately predict human–object interactions and outperform 3D convolution networks in recognising daily activities.

To use TRN to refine frame-level label prediction, we provide the concept of an anchor based on the study of [26]:

1. An anchor is a video clip
2. Each anchor has a different number of video frames, so each anchor represents a video clip with a different time scale
3. All anchors take \(x_i\) as the ending time

We feed all anchors \((a_1, a_2, a_3, ...\) into the ORR, and each will generate an output, \(\overrightarrow{s}_{1}, \overrightarrow{s}_{2}, \overrightarrow{s}_{3}, ...\). The output is used to refine frame-level label prediction \(y_i\), as shown in Figure 4 and Algorithm 3.

5 | EXPERIMENTS

To evaluate the proposed online dense human activity detection method, we conducted a set of experiments, testing its online performance during dense activity detection, and we compare the proposed method with state-of-the-art offline methods on activity detection datasets, including Charades [19] andAVA [27].

5.1 | Experimental setup

5.1.1 | Datasets

The Charades dataset is a human activity dataset with multiplicity and diversity. It contains 9848 videos across 157 human daily activities, such as preparing a meal and opening a laptop. More important, each video contains 6.8 activities on average and the activities often overlap temporally, so the need to evaluate dense activity detection is satisfied.

The atomic visual actions (AVA) dataset was published by Google Inc. It densely annotates 80 atomic visual actions in 192 video clips. The AVA dataset contains realistic scenes and complex human activities, which makes it challenging. It provides not only frame-level activity labels but also location labels within each frame. Here, we use only frame-level label data for to evaluate the algorithm.
Algorithm 3

Input: Input stream \( \{x_i\}_{i=1}^L \), the coarse label prediction module \( \{\hat{y}_i\}_{i=1}^L \).
Output: Each activity possibilities at frame-level refined by HOI module \( \{y_i\}_{i=1}^L \).
1 for each frame in sliding window \( W_i \) do
2 Perform Faster RCNN to obtain bounding boxes of instances: \( b_h \) for a person and \( b_o \) for an object;
3 Get bounding box and corresponding score based on instance appearance \( \{b_h, s_h\}, \{b_o, s_o\} \);
4 Get activity score for activity \( c \) based on appearance instances \( s_h, s_o \);
5 Get attention score \( s_{act} \) for activity \( c \) based on human-object pairs \( \{b_h, b_o\} \);
6 Get the activity score by Equation 9:
7 The refined frame-level results after HOI module
8 \[ \bar{y}_i = \left( \frac{y_i^1 + y_i^c}{2} \right) \]
9 end

Algorithm 4

Input: Input stream \( \{x_i\}_{i=1}^L \), the frame-level detection results refined by HOI module \( \{y_i\}_{i=1}^L \).
Output: Each activity possibilities at frame-level \( \{\hat{y}_i\}_{i=1}^L \) for predefined activity classes.
1 Initialize According to the position of the sliding window \( W_i \), a set of \( l \) anchors \( \{a_k\}_{k=1}^l \) of length \( N, 2N, \ldots, l \cdot N \) respectively are constructed.
2 for each anchor in \( \{a_k\}_{k=1}^l \) do
3 Select different number of frames evenly from the current anchor, which represents the current anchor input into the ORR module;
4 Get the segment-level activity prediction results of each anchor \( \bar{s}_k \) = \( \{s_1^k, s_2^k, \ldots, s_l^k\} \);
5 end
6 for each frame \( x_i \) do
7 for each activity class \( c \) do
8 for each segment-level result of each anchor \( \bar{s}_k \) do
9 Get the refined frame-level results of \( x_i \) for activity \( c \) from \( y_i^c \);
10 Get segment-level results of \( x_i \) for activity \( c \) from \( s_k^c \);
11 \( y_i^c = \bar{s}_k^c + y_i^c \);
12 end
13 end
14 end

Statistics of dense activities on Charades and AVA are shown in Table 1. For Charades, we found that up to 19 labels are assigned to one frame. In addition, 14.56% of frames in the training set have more than four labels, and 28.44% of frames in the test set. In AVA, up to nine labels are assigned to one frame. Different from AVA, the dataset contains about half of frames with no label. In the training set, 7.08% of frames have more than four labels; in the test set, the percentage is 11.38%.

5.1.2 Implementation details

Software: Our code is based on Pytorch and Tensorflow. Image features in aggregated-event generation and coarse label detection are calculated by I3D [17], which is a two-stream 3D CNN that achieved state-of-the-art performance on several action recognition tasks [1]. In the HOI part, we use Faster R-CNN [23] and ResNet-50 [18] to generate human and object bounding boxes. The object detection part is completed by open Detectron [28], version 1.

Hardware: In the training stage, our method requires massive computation, so our algorithm is designed based on a GPU platform. We use Tesla K80 graphics processing unit (GPU) to train our networks. Tesla K80 has two GPUs and a 4992 NVIDIA CUDA core with 24 GB of GDDR5 memory and 480 GB/s aggregate memory bandwidth.

5.2 Evaluation metrics

Mean average precision: Mean average precision (mAP) is an evaluation criterion for multilabel image classification algorithms, in which it is inappropriate to measure its performance simply by criteria such precision and recall, because one image might have several labels. mAP is a commonly used criterion for activity detection in video; it can simultaneously measure precision and recall of an algorithm over multiple classes [1].

Precision of dense activity detection \( R_{d_{n_{a}}}^{(s_{a})} \): \( R_{d_{n_{a}}}^{(s_{a})} \) is designed to measure performance when detecting dense activities at the frame level. \( n_{d} \) is the number of dense activities that can be properly detected. To calculate \( R_{d_{n_{a}}}^{(s_{a})} \), we check all of the video frames with more than \( n_{d} \) labels. If \( n_{d} \) labelled activities for one frame are detected, the detection of the frame is correct. \( R_{d_{n_{a}}}^{(s_{a})} \) is the ratio between the number of correctly detected frames and the total number of the frames with more than \( n_{d} \) labels. We let \( R_{d_{n_{a}}}^{(s_{a})} = \infty \) represent the case in which all labelled activities have to be correctly detected.

5.3 Online dense activity detection

An example of dense activity detection process is shown in Figure 5. The sample is from the Charades dataset, and the
| Charades (training set) | Charades (test set) | AVA (training set) | AVA (test set) |
|------------------------|---------------------|--------------------|----------------|
| 0–10.98%               | 0–7.83%             | 0–47.94%           | 0–47.19%       |
| 1–18.61%               | 1–10.26%            | 1–17.52%           | 1–17.95%       |
| 2–22.83%               | 2–14.95%            | 2–9.55%            | 2–8.99%        |
| 3–19.84%               | 3–16.74%            | 3–9.14%            | 3–9.68%        |
| 4–13.18%               | 4–16.57%            | 4–8.77%            | 4–4.81%        |
| >4%–14.56%             | >4–28.44%           | >4–7.08%           | >4–11.38%      |

Abbreviation: AVA, atomic visual actions.
The statistical data are listed in the format of n/p, where n stands for how many activities are labelled for one frame and p% is the percentage of frames with n activity labels.

**TABLE 1** Statistics of dense activities on Charades and AVA

**FIGURE 5** Example of online dense activity detection. The video clip is a sample from the Charades dataset with ID N93NK. Up to four activities are detected by our algorithm at frame level.

**FIGURE 6** Curves of confidence score for different activities of video N93NK in the Charades dataset.

The video is about a person interacting with a book. At the beginning of the video, our algorithm found the activity ‘holding a book’. From the 25th to 110th frame, the person is out of the image and no activity is detected. From the 160th to 340th frame, our algorithm found four activities: holding a book, putting a book somewhere, smiling at a book, and putting something on table. We observed that the activities have different temporal durations, and sometimes one activity might overlap another. The curves of confidence scores for different activity classes are shown in Figure 6.

### 5.4 Comparison with offline methods

In this section, we mainly compare our method with two state-of-the-art methods: the baseline method [17] and the super-event method [1]. The base feature extraction component of the two methods is a CNN model called I3D [17]. I3D is a variant of the two-stream method [15], which can take an RGB image, optical flow data or both an RGB image and optical (two-stream) as its input. Thus, both the baseline and super-event methods have three combinations: RGB (using RGB image data), flow (using optical flow data) and two-stream (using both an RGB image and optical flow data). In our method, we use only RGB images as the input. The code for the baseline method is provided by Piergiovanni et al. [1]; it is a pretrained, non–fine tuned model.
FIGURE 7 Incremental input scheme

FIGURE 8 Comparison with offline methods on Charades dataset and atomic visual actions (AVA) dataset. The online methods are the baseline method [17] and superevent method [1]. They both adopt inflated 3D ConvNet as their base detection component, which can take RGB images, optical flow images or both as input.

TABLE 2 Frame–mean average precision on Charades dataset when 100% video is observed

| Methods                        | Mean average precision, % |
|--------------------------------|---------------------------|
| RGB [18]                        | 7.89                      |
| Two-stream [18]                 | 8.94                      |
| R-C3D [14]                      | 12.7                      |
| Single-stream R-C3D plus OHEM [20] | 13.3                      |
| Fine-tuned I3D [17]             | 17.22                     |
| Temporal aggregation network [3] | 17.6                      |
| I3D plus long short-term memory  | 18.1                      |
| I3D plus stacked spatiotemporal graph convolutional network [20] | 19.09 | |
| I3D plus superevents [1]        | 19.41                     |
| I3D plus 3TGMs plus superevents [30] | 21.8                      |
| ST-MPNN plus Symb plus I3D plus biGRU [31] | 23.4                     |
| Proposed method                 | 25.34                     |

ABBREVIATIONS: OHEM, online hard example mining; ST-MPNN, Spatio Temporal Message Passing Neural Network; Symb, Symbolic Graph; 3TGM, 3 Temporal Gaussian Mixture layers; I3D, inflated 3D ConvNet.

TABLE 3 Frame–mean average precision on AVA dataset when 100% video is observed

| Methods                        | mAP, % |
|--------------------------------|--------|
| Random                         | 2.65   |
| I3D RGB                        | 8.3    |
| I3D flow                       | 9.1    |
| I3D plus superevents [1]       | 9.8    |
| Proposed method                | 11.2   |

ABBREVIATIONS: AVA, atomic visual actions; I3D, inflated 3D ConvNet.

For comparison with offline methods, we adopt an incremental way to generate the input video frames. Video clips are divided into 20 smaller clips, and each time, a 5% incremental part is added to the input, as illustrated in Figure 7.

The performance (mAP) of the state-of-the-art offline methods and our method on Charades and AVA are shown in Figure 8. Our method outperforms the two offline methods by a significant margin, improving mAP by about 5%. We also observed that the curves of superevent methods have an obvious decline at the end. We believe that this is because the proposed window sampling strategy better summarises the historical frames than simply inputting the whole video clip.

When video observation reaches 100%, the comparison of the frame-mAP of existing dense activity detection methods is shown in Tables 2 and 3. The proposed method uses a window sampling strategy in the video observation process.

5.5 Dense activity detection

In this section, we conducted more experiments to evaluate the performance of dense activity detection with the precision of dense activity detection ($R_{du}$). The experimental results are listed in Tables 4 and 5. When $\eta_d = 1$, the methods are actually working in the single activity detection model and the mAP
values are relatively high. The mAP values show an obvious drop-off with increasing $R(n,d)$. Our method had better performance than the two offline methods on both the Charades and AVA datasets.

5.6 Computational complexity

Computational complexity is an important criterion to measure the performance of online methods. In this section, we evaluate the proposed window sampling strategy.

In the experiment, we use our method and the superevent method to evaluate the window sampling strategy with the incremental input scheme (in Section 5.4). Each method is fed with two kind of inputs: video frames that increase by 5% of the whole video each time and video frames after window sampling ($N$ frames).

We present the experimental result in Figure 9. The computational complexity of methods with window sampling equals the length of the aggregated event window ($N = 32$) and remains unchanged after more increments are added. However, the computational complexity of the methods without the window sampling strategy will be linear proportional to the numbers of input video frames. We also observed no obvious difference in performance (mAP) between the methods with window sampling and those without window sampling. Thus, the window sampling strategy can greatly reduce computational complexity while it keeps performing during dense activity detection.

6 | CONCLUSION

We propose a two-stage online framework of dense human activity detection. To improve detection, we divided the detection process into two stages: warm-up and detection. A context model called OAE can be generated during the warm-up stage. The detection stage is made up of two modules: coarse label prediction and refined label prediction. The OAE model provides an a priori context in coarse label prediction. Then, we provides HOI detection and ORR to refine coarse label prediction at the frame level. Compared with several well-performing offline methods, we are able to confirm that the proposed method is significantly better in online dense activity detection compared with dense activity datasets, the Charades and AVA datasets.

ACKNOWLEDGEMENTS

This work was supported by the Programme for Innovative Research Team at the University of Tianjin (No. TD13-5032) and the Tianjin Science and Technology Programme (No. 19PTZWHZ00020).

REFERENCES

1. Piergiovanni, A., Ryoo, M.S.: Learning latent super-events to detect multiple activities in videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5304–5313. Salt Lake City (2018)
2. Yeung, S., et al.: Every moment counts: dense detailed labelling of actions in complex videos. Int J Comput Vis. 126(2-4), 1–15 (2018)
3. Dai, X., et al.: Tan: Temporal aggregation network for dense multi-label action recognition. IEEE Conference on Winter Conference on Applications of Computer Vision. Waikoloa Village (2018)
4. Nauata Junior, N.I., et al.: Structured label inference for visual understanding. IEEE Trans Pattern Anal Mach Intell, 1 (2019)
5. Zhao, Y., et al.: Temporal action detection with structured segment networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2914–2923. Venice (2017)
6. Shou, Z., et al.: Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5734–5743. Honolulu (2017)
7. De Geest, R., et al.: Online Action Detection. Computer Vision-ECCV, pp. 269–284. Amsterdam (2016)
8. Ma, S., et al.: Learning activity progression in lstms for activity detection and early detection. In: Ieee Conference on Computer Vision and Pattern Recognition, pp. 1942–1950 (2016)
9. Gao, J., et al.: RED: Reinforced Encoder-Decoder Networks for Action Anticipation. arXiv preprint arXiv:1707.04818. 28th British Machine Vision Conference, (BMVC), London (2017)
10. Soomro, K., et al.: Online localization and prediction of actions and interactions. IEEE Trans Pattern Anal Mach Intell. 41(02), 459–472 (2019)
11. Gao, J., et al.: Turn tap: temporal unit regression network for temporal action proposals. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 3628–3636 (2017)
12. Gao, J., et al.: Cascaded Boundary Regression for Temporal Action Detection. British Machine Vision Conf. (BMVC), London (2017)
13. Dai, X., et al.: Temporal context network for activity localization in videos. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5793–5802 (2017)
14. Xu, H., et al.: R-c3d: region convolutional 3d network for temporal activity detection. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5783–5792 (2017)
15. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in video. Adv Neural Inf Process Syst. 24, 1–9 (2014)
16. Tran, D., et al.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, pp. 4489–4497. December (2015)
17. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308. Honolulu (2017)
18. Sigurdsson, G.A., Divvala, S., Farhadi, A.: Asynchronous temporal fields for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 585–594. Honolulu (2017)
19. Sigurdsson, G.A., et al.: Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. Computer Vision-ECCV, pp. 510–526. Amsterdam (2016)
20. Ghosh, P., et al.: Stacked Spatio-Temporal Graph Convolutional Networks for Action Segmentation. arXiv preprint arXiv:1811.10575. Snowmass (2020)
21. Gregor, K., et al.: DRAW: a recurrent neural network for image generation. In: Proceedings of the 32nd International Conference on Machine Learning (PMLR), 1462–1471 (2015)
22. Gao, C., et al.: Ican: instance-centric attention network for human-object interaction detection. In: British Machine vision Conf (BMVC) (2018)
23. Ren, S., et al.: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 39(6), 1137–1149 (2017)
24. Chao, Y.W., et al.: HICO: a benchmark for recognising human-object interactions in images. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 1017–1025. Santiago (2015)
25. Zhou, B., et al.: Temporal Relational Reasoning in Videos, Computer Vision-ECCV, pp. 803–818. Munich (2018)
26. Lin, T.Y., et al.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 936–944. Honolulu (2017)
27. Gu, C., et al.: Avac a video dataset of spatio-temporally localised atomic visual actions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6047–6056. Salt Lake City (2018)
28. Girshick, R., et al.: Detectron. Facebook. https://github.com/facebookresearch/detectron. Accessed 2018
29. Xu, H., et al.: Two-Stream Region Convolutional 3d Network for Temporal Activity Detection. IEEE Trans Pattern Anal Mach Intell. 41(10), 2319–2332 (2019)
30. Piergiovanni, A.J., Fan, C., Ryoo, M.S.: Learning latent subevents in activity videos using temporal attention filters. In: Proceedings of the Thirty-first AAAI Conference on Artificial Intelligence, pp. 4247–4254. San Francisco (2017)
31. Mavroidi, E., et al.: Neural Message Passing on Hybrid Spatio-Temporal Visual and Symbolic Graphs for Video Unstanding. arXiv preprint arXiv:1905.07385. Glasgow (2019)