**Supplementary Materials**

**Supplementary Methods**

**Preprocessing**

We implemented spectral subtraction1 for noise reduction, downsampled the raw audio data from 44.1kHz to 8.82kHz and split it into 5-minute segments. Next, each segment was transformed into Mel spectrogram representation, serving as the input to our deep learning model. A Mel spectrogram is a visual depiction that details a signal’s frequency composition over time. Several studies has confirmed that Mel spectrogram provided an efficient and perceptually relevant representation and performed well in various tasks, while being of smaller size with reduced computation in training and prediction2. In this study, a Mel spectrogram was computed using a hop size of 512 samples (58ms) with a 2048-point discrete Fourier transform aggregated to yield 64 Mel bins per frame. Consequently, we were able to locate any apneic event detected from the Mel spectrogram in the corresponding raw audio data.

**OSAnet construction**

Traditional approaches have relied on manually selected acoustic features or other human-engineered features related to apnea. Such methods may be influenced by subjectively selected variables. On the contrary, we aimed to develop an intelligent diagnostic approach that could automatically extract relevant information from snoring sound for apnea detection. In the past few decades, deep learning has achieved remarkable success and has shown strong ability in various computer vision tasks.3,4 In our study, sleep apnea detection was considered an object detection problem and convolutional neural networks were adopted for the task, since they allowed us to stack more layers to detect higher-level concepts, while still being invariant to translation.4 The deep convolutional neural networks we proposed was named OSAnet, which not only learned to extract apnea-related features from the audio input but located and classified the apneic events based on these features.

OSAnet consisted of feature extraction layers and feature fusion layers (Figure S1). In feature extraction layers, we reduced the dimension of input data from 64 to 1 in dimension reduction layers and ResNet50-like backbone networks were used to extract features.5 The modification of ResNet50-like networks we introduced was to replace the two-dimensional convolutional layers and pooling layers with one-dimensional ones. The shallow layers mainly extracted low-level features while the deeper layers distilled the high-level semantic features based on low-level features.

Following the feature extraction layers, we added three extra feature layers for feature fusion and apneic event detection. It is noteworthy that we not only produced predictions at multiple layers but built a top-down and bottom-up pyramidal feature hierarchy6 to combine the detailed information from the low-level layers (smaller receptive field sizes) with the semantic information from the high layers (larger receptive field sizes). The high-level features of low-resolution, high-semantic information and the low-level features of high-resolution, low-semantic information were connected from top to bottom, so that the features at all scales involved rich semantic information. By making predictions at multiple layers, we managed to handle apneic events of different durations. With a low-resolution feature, we upsampled the spatial resolution of high-level features by a factor of two (using one-dimensional transposed convolution). The upsampled features were then merged with the corresponding bottom-up features by element-wise addition, which underwent a 1×1 convolutional layer to change dimensions. This process was iterated until the highest resolution feature is generated. Finally, with the help of a series of convolutional filters, these layers were successful to produce a fixed set of detection predictions.

**Prior bounding boxes**

According to AASM 2012 scoring criteria, the duration of sleep apneic and hypopneic events is supposed to be more than 10 seconds. To cover various durations of sleep events in different feature maps with different receptive field sizes, we selected multiple durations as the base lengths of each feature map and then scaled up the base lengths with a series of aspect ratios to obtain a set of intervals. These intervals were named prior bounding boxes, as used in SSD (Figure S2)7. Each feature map contained numerous locations where the prior bounding boxes could locate. The statistical distribution of apneic event duration was incorporated into the durations of feature maps to design a series of scaling factors, which was employed to compute the base lengths. Supposing there were feature maps for prediction, the scaling factor of each feature map was generated as:

where was 0.033 and was 0.133, meaning the lowest-level layer had a scale of 0.033 and the highest-level layer had a scale of 0.133, and all layers in between were regularly spaced. They were given based on the ratio of sleep apneic events to the total duration of input data (5 minutes). Since the duration of input data was 5 minutes, the base lengths of Feature Map 1 to Feature Map 4 were set to 10 seconds, 20 seconds, 30 seconds, 40 seconds. Then we imposed different aspect ratios for the prior bounding boxes and denoted them as (for better demonstration, only 4 aspect ratios are depicted in Figure S2)**.** Then we computed the duration for each prior bounding box and thereby resulted in 8 boxes per feature map location. We set the center of each prior bounding box to, where was the duration of -th feature map, .

The feature map was tiled in a convolutional manner by these prior bounding boxes, so that the location of each prior bounding box relative to the feature map was fixed. For each box, two offsets, viz. center offset () and duration offset () were predicted. The confidence loss was also calculated, implying the confidence score to which the prior bounding box belonged. As there were 8 prior bounding boxes, this resulted in a total of  filters that were applied for each location in the feature map. Specifically, the number of classes is 2, including sleep events and background noise. After that, a predicted box was derived from the prior bounding box adjusted by the two offsets, indicating an instance of apneic event was predicted by OSAnet.

**Training**

The model was trained in an end-to-end manner, which has been verified as an effective method in multiple tasks including computer vision. As there were 8 prior bounding boxes at every feature map location and the durations of feature map could be extremely large, it was vital to enhance computational efficiency. Therefore, we deployed the following matching strategy to identify which prior bounding box corresponded to the ground truth (an apneic event). We matched each ground truth to the prior bounding boxes and select those with the best Jaccard overlap. The prior bounding boxes matched to any ground truth with Jaccard overlap higher than a threshold (0.5) were considered as matched ones as well. The training objective was derived from SSD, but was modified to handle one-dimensional data. Let be an indicator for matching the -th prior bounding box to the -th ground truth of class (Here, the number of classes is 2, including apneic events and background noise). Following the matching strategy above, we were able to ensure . The total objective loss was divided into two parts, the localization loss () and the confidence loss (),

where  was a term to balance the weight of and , in this experiment ,  is the number of matched prior bounding boxes and if = 0, we set the loss to 0. The localization loss is a Smooth loss between the predicted box and the ground truth box . We regressed to the offsets for center and duration of the prior bounding boxes .

The confidence loss is the softmax loss over multiple classes confidences .

where

After the matching step, we noticed the number of negatives was considerably larger than the positives, leading to a significant imbalance between the positive and negative training examples. Thus, an approach named hard negative mining was introduced, in which we adjusted the ratio between the negatives and positives to be at most 3:1 through sorting them using the highest confidence loss for each prior bounding box when selecting negative examples and picking the top ones. This process helped to optimize and accelerate the training stably.

**Inference**

After the OSAnet was trained, in order to locate the position of apneic events in 5-minute segment, we designed One-Dimensional Non-Maximum Suppression (1D-NMS) method with reference to NMS8, a classic post-processing method in object detection. The details are illustrated in **Algorithm S1**, where *P* is denoted as the list of prior bounding boxes whose confidence scores exceed detection threshold , *S* as the corresponding confidence scores, *predApena* as the predicted apneic event. The prior bounding box with maximum confidence score *m* is denoted as . If the prior bounding boxes with lower confidence scores overlaps with , will be removed from *P,* and the corresponding confidence scores will be removed from *S.* Finally, we add to *predApena.* The above process will be repeated until there are no prior bounding boxes left in *P*.

|  |
| --- |
| **Algorithm S1** One-Dimensional Non-Maximum Suppression |
| **Input:** *P={*,,,}, *S*={,,,}  **Output:** *predApena*  2: *predApena* = []  3: **while** *P empty* **do**  4: *m* = argmax(*S*)  5: **for** **in** *P* **do**  6: **if** IoU(, ) 0 **then**  7: *P.*remove()*; S.*remove()  8: **end**  9: *predApena* *= predApena* *.*append()*;*  10: **end**  11: **end**  12: **return** *predApena* |

**The Detecting While Slicing method**

A critical issue we noticed was that the 5-minute segments could include some apneic events which had been cut in half during preprocessing, and it would inevitably undermine the accuracy of OSAnet. That being the case, we developed a novel method named Detecting While Slicing **(Algorithm S2)**, which was proposed according to two principals: the durations of most apneic events were no more than two minutes; the intervals between two apneic events were not less than ten seconds. The Detecting While Slicing method denotes that the duration after each detected apneic event in the present segment may involve an incomplete event, and that duration is supposed to be detected again in the next segment. The algorithm of Detecting While Slicing is shown below.

Let *len(audio)* be the duration of whole night snoring sound data and define the function respEventDetector as the OSAnet algorithm. The input and output of the function respEventDetector is a 5-minute segment and a predicted respiratory event, respectively. The predicted respiratory event is presented by *predApnea* = [[*predApnea1 o, predApnea1 e*],…,[[*predApneak o, predApneak e*], where *predApneak o* and *predApneak e*denote the onset and end of the k-th predicted respiratory events. When the pointer *p* starting from 0, *Audio*[*p* : *p*  300] (denoting the duration of 300 seconds following the pointer *p*) is input and the location of predicted respiratory event is computed and saved into *eventList*. If the maximum of end time*, predApneamax* is larger than 180 s, we set *p* as *p* *predApneamax* 10, otherwise as *p* 180.

|  |
| --- |
| **Algorithm S2** Detecting While Slicing |
| **Input:** sleep sound signal of participant, *audio*;  **Output:** respiratory event list of participants, *eventList*;  1: *p* = 0  2: *eventList* = []  3: **while** *plen*(*audio*)**do**  4: *predApnea*= respEventDetector (*audio*[*p* : *p*  300])  5: add *predApnea* to *eventList*  6: // Save predicted respiratory events of present segment;  7: // *predApnea*= [[*predApnea1 o, predApnea1 e*],…,[[*predApneak o, predApneak e*]  8: *predApneamax* = max(*predApnea1 e*,…, *predApneak e*]  9: **if** *predApneamax*180 **then**  10: *p* = *p* *predApneamax* 10  11: // Next segment starting from *p* *predApneamax* 10  12: **else**  13:  *p* = *p* 180  14: // Next segment starting from *p* 180  15: **end if**  16: **end while**  17: **return** *eventList;* |

**Table S1**

The Configuration of Dimension Reduction Layers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layers | Number of Filters | Filter Size | Stride | Padding |
| Conv | 16 | 3×3 | 1×1 | 1×1 |
| Conv | 16 | 3×3 | 1×1 | 1×1 |
| pool | 16 | 4×4 | 4×1 | 0×0 |
| Conv | 32 | 3×3 | 1×1 | 1×1 |
| Conv | 32 | 3×3 | 1×1 | 1×1 |
| pool | 32 | 4×4 | 4×1 | 0×0 |
| Conv | 64 | 3×3 | 1×1 | 1×1 |
| Conv | 64 | 3×3 | 1×1 | 1×1 |
| pool | 64 | 4×4 | 4×1 | 0×0 |

The Configuration of Extra Feature Layers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Blocks | Layers | Number of Filters | Filter Size | Stride | Padding |
| Extra Feature Layer 1 | Conv | 512 | 1×1 | 1×1 | 0×0 |
| Conv | 256 | 1×2 | 1×2 | 0×0 |
| Extra Feature Layer 2 | Conv | 128 | 1×1 | 1×1 | 0×0 |
| Conv | 256 | 1×2 | 1×2 | 0×0 |
| Extra Feature Layer 3 | Conv | 128 | 1×1 | 1×1 | 0×0 |
| Conv | 256 | 1×2 | 1×2 | 0×0 |



Figure S1-The architecture of OSAnet



At each location of a feature map, OSAnet generated a set of prior bounding boxes with a base length and 8 aspect ratios. Here, only 4 aspect ratios are depicted for clarity.

Figure S2-Generation of prior bounding box

**References:**

1. Berouti M, Schwartz R, Makhoul J. Enhancement of speech corrupted by acoustic noise. ICASSP'79 IEEE International Conference on Acoustics, Speech, and Signal Processing; 1979: IEEE. p. 208-11.

2. Choi K, Fazekas G, Sandler M, Cho K. A comparison of audio signal preprocessing methods for deep neural networks on music tagging. 2018 26th European Signal Processing Conference (EUSIPCO); 2018: IEEE. p. 1870-4.

3. Aytar Y, Vondrick C, Torralba A. SoundNet: Learning Sound Representations from Unlabeled Video. 2016.

4. Sainath TN, Kingsbury B, Saon G, et al. Deep Convolutional Neural Networks for Large-scale Speech Tasks. Neural Networks 2015;64:39-48.

5. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 27-30 June 2016. p. 770-8.

6. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature Pyramid Networks for Object Detection. 2016.

7. Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector. European Conference on Computer Vision; 2016.

8. Neubeck A, Gool LV. Efficient Non-Maximum Suppression. 18th International Conference on Pattern Recognition (ICPR'06); 2006 20-24 Aug. 2006. p. 850-5.