Count What You Want: Exemplar Identification and Few-Shot Counting of Human Actions in the Wild

Yifeng Huang[1*], Duc Duy Nguyen[2*], Lam Nguyen[2], Cuong Pham[2,3], Minh Hoai[1,2]

[1]Stony Brook University, NY, USA
[2]VinAI, Hanoi, Vietnam
[3]Posts & Telecommunications Institute of Technology, Hanoi, Vietnam

Abstract

This paper addresses the task of counting human actions of interest using sensor data from wearable devices. We propose a novel exemplar-based framework, allowing users to provide exemplars of the actions they want to count by vocalizing predefined sounds "one", "two", and "three". Our method first localizes temporal positions of these utterances from the audio sequence. These positions serve as the basis for identifying exemplars representing the action class of interest. A similarity map is then computed between the exemplars and the entire sensor data sequence, which is further fed into a density estimation module to generate a sequence of estimated density values. Summing these density values provides the final count. To develop and evaluate our approach, we introduce a diverse and realistic dataset consisting of real-world data from 37 subjects and 50 action categories, encompassing both sensor and audio data. The experiments on this dataset demonstrate the viability of the proposed method in counting instances of actions from new classes and subjects that were not part of the training data. On average, the discrepancy between the predicted count and the ground truth value is 7.47, significantly lower than the errors of the frequency-based and transformer-based methods. Our project, code and dataset can be found at https://github.com/cvlab-stonybrook/ExRAC.

Introduction

Counting human actions of interest using wearable devices is a crucial task with applications in health monitoring (e.g., Baghdadi et al. (2021)) and performance evaluation (e.g., O'Reilly et al. (2018)). However, the majority of existing counters are often designed for a limited set of action categories, such as walking and a few other physical exercises. These class-specific counters (e.g., Genovese, Mannini, and Sabatini (2017)) are incapable of handling classes beyond those they have been explicitly trained for. Consequently, relying solely on class-specific counters becomes impractical and unscalable when dealing with a diverse set of action categories. For scalability, a promising alternative to class-specific counters is class-agnostic counters, capable of tallying repetitions from any arbitrary class, as long as this class represents the dominant activity within the sensor data being analyzed.

However, in many real-world scenarios, our interest might not lie in counting actions from the dominant class. For instance, in sports training and skill evaluation, the objective is often to detect specific and infrequent mistakes within the prevalent data. As illustrated in Fig. 1, the action of interest may occur only briefly within the entire data sequence. These factors pose significant challenges when applying existing methods effectively.

Confronting the challenge presented by real-world data, which often contains undesired actions, we propose to develop an exemplar-based counting method, where an user can provide exemplars of what they want to count. However, the development of such a method poses two significant technical challenges. Firstly, devising a convenient exemplar provision scheme is nontrivial. Secondly, once we have some exemplars, the question remains how to effectively leverage them. In this paper, we address both of these challenges to develop a novel exemplar-based counting method.

For the first challenge, we propose an intuitive and non-intrusive approach for specifying exemplars using vocal sounds. The exemplars are conveniently provided by verbally counting out loud "one," "two," "three" at the onset of the counting process as shown in Fig. 1. Each utterance corresponds to one repetition. To accurately detect the positions of these counting utterances in the audio sequence, we develop an efficient algorithm that solves a constrained optimization problem with the two constraints on the temporal ordering and the temporal distance between the identified positions. Once the positions of the counting utterances are identified, we extract the exemplars from these locations.

For the second challenge, we propose a novel model that jointly processes the exemplars with the whole data sequence as shown in Fig. 1. More concretely, we first generate per-window embeddings for both the exemplars and the whole data sequence. Subsequently, we compute a similarity map between the exemplar and data sequence embeddings, using Soft-DTW (Cuturi and Blondel 2017) and correlation measures. This similarity map serves as the basis for generating a sequence of exemplar-infused embeddings for the data sequence. The initial embedding sequence and the exemplar-infused embedding sequence are then fed into a density estimation module for moment-by-moment density estimation, from which the final count is obtained by summing the density values.

*These authors contributed equally to this work.

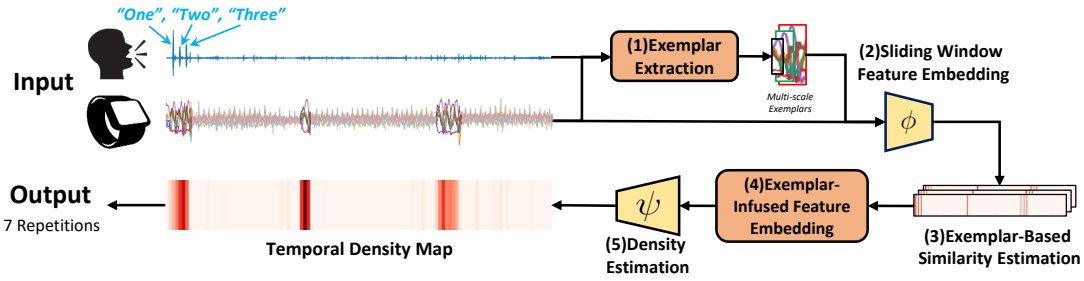


Figure 1: Processing pipeline of our method. The input consists of the sensor signal and the audio signal containing the utterances "one," "two," and "three," corresponding to three repetitions of the action of interest. The output is the total count, obtained by summing the values of the intermediate 1D density profile. This profile is better visualized as a 2D map as shown here. This figure also shows the other processing steps, which will be explained in the forthcoming method section.

Realizing the importance of a good similarity measurement, we introduce a novel distance-preserving loss. This loss enforces the high-dimensional per-window embeddings to maintain local patterns, thereby preserving the similarity relationships observable in the lower-dimensional space. In addition, considering the limited training data, we propose an exemplar-based data synthesis pipeline, which can synthesize training data and improve the result significantly.

To develop and evaluate the proposed method, we have collected a dataset named **D**iverse **W**earable **C**ounting dataset (**DWC**). This dataset comprises sensor data sequences accompanied by audio-specified exemplars collected from 37 subjects performing 50 distinct action categories. What sets this dataset apart from many existing ones is the availability of synchronized audio data with vocal sounds for specifying exemplars. Furthermore, this dataset includes instances where the action of interest may not be the predominant action within the data sequence, providing a more realistic representation of real-world scenarios.

In short, the main contributions of our paper are threefold. First, we introduce a novel strategy for using audio to specify exemplars of what needs to be counted. Second, we propose a novel counting method that utilizes exemplars, incorporating a distance-preserving loss and an exemplar-based data synthesis pipeline. Third, we introduce an unique dataset with multiple data modalities to develop a practical counting method for real-world scenarios.

Related Work

Action counting through wearable devices is driven by its diverse range of applications in health monitoring (Baghdadi et al. 2021; Lee et al. 2015; Nam, Kim, and Lee 2016; Hatamie et al. 2020; Ramachandran and Liao 2022; Patel et al. 2010), sports training (Chang, Chen, and Canny 2007; O'Reilly et al. 2018; Kranz et al. 2013; Ding et al. 2015), and industrial contexts (Kong et al. 2019; Stiefmeier et al. 2008). Existing counting methodologies have predominantly focused on particular action categories, such as physical exercises (Genovese, Mannini, and Sabatini 2017; Kupke et al. 2016; Pillai et al. 2020; Bian et al. 2019; Ishii et al. 2021; Morris et al. 2014; Soro et al. 2019a; Oh, Olsen, and Ramamurthy 2020). This specialization restricts their adaptability, especially when faced with classes having no prior training data. Consequently, relying on class-specific counters proves inadequate and unscalable in managing the wide range of action categories encountered in real world.

Class-agnostic counters is an alternative to class-specific counters, but they can only count repetitions from the dominant class. Earlier strategies, based on Fourier analysis or wavelet transforms (Cutler and Davis 2000; Azy and Ahuja 2008; Pogalin, Smeulders, and Thean 2008; Runia, Snoek, and Smeulders 2018), peak detection (Thangali and Sclaroff 2005), and singular value decomposition (Chetverikov and Fazekas 2006), have been explored. More recently, significant attention has been directed towards repetitive action counting in videos (Levy and Wolf 2015; Zhang et al. 2020; Zhang, Shao, and Snoek 2021; Fieraru et al. 2021; Hsu et al. 2021; Hu et al. 2022; Dwibedi et al. 2020). Recent works (Dwibedi et al. 2020; Hu et al. 2022) have achieved promising results by harnessing temporal self-similarity to count repetitive actions from the dominant class.

While exemplar-based counting is not a novel concept, our contribution stands as one of the few approaches designed for wearable devices. Notably, it marks the pioneering effort in introducing a strategy for specifying exemplars through the act of uttering and subsequently detecting predefined vocal sounds. This approach is innovative and distinct from existing works in various fields. For instance, in computer vision, there are methods that utilize exemplars for counting objects in images (Liu et al. 2022; Yang et al. 2021; Ranjan et al. 2021; Ranjan and Hoai 2022b; Shi et al. 2022; Lu, Xie, and Zisserman 2018; You et al. 2023; Nguyen et al. 2022; Huang, Ranjan, and Hoai 2023; Ranjan and Hoai 2022a). These methods require users to specify exemplars by drawing bounding boxes. However, when dealing with time-series data, the natural provision of exemplars becomes non-trivial. First, the visualization and semantic parsing of sensor data pose greater challenges compared to images. Second, manually determining the temporal extents of human actions in time series is more difficult compared to delineating object bounding boxes in images. Third, for sensor-based counting, immediate results are often required, making it crucial for the process of providing and identifying exemplars to be convenient and efficient, without involving time-consuming procedures such as transmitting, visualizing, and drawing.

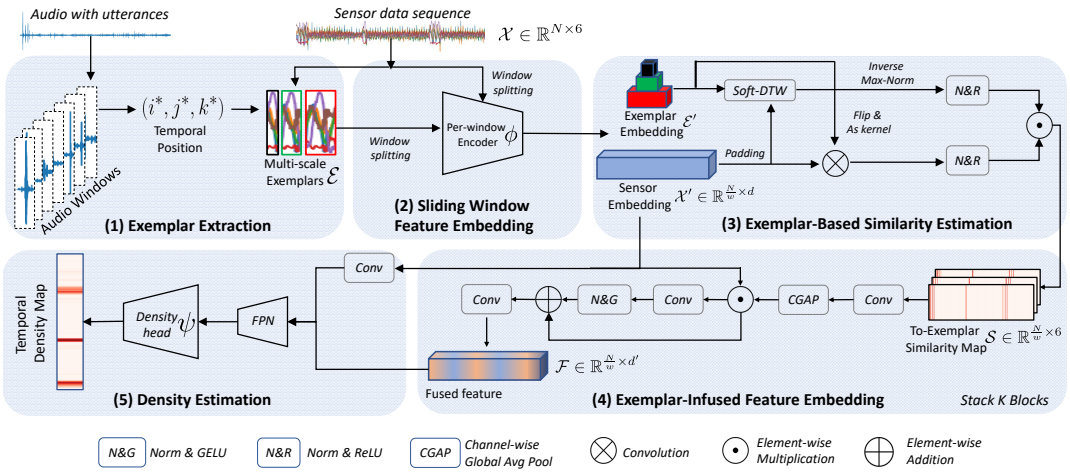


Figure 2: Main steps of our method. Our method begins with exemplar extraction, which is based on predefined utterance detection in the audio data. Following this, per-window embeddings are extracted. Subsequently, we compute the similarity between the entire sensor sequence and the exemplars, which is then used for feature fusion. Finally, the temporal density map is estimated based on the fused features and the sensor embeddings.

Proposed Approach

Our objective involves tallying the occurrences of a specific action class within a sequence of sensor data. Our method takes as input both the sensor data sequence and an audio sequence synchronized with it, featuring predetermined vocal sounds – one, two, three – corresponding to the initial three repetitions of the action. As such, our approach comprises two fundamental stages: first, the identification of exemplars, and subsequently, their utilization to derive the overall count. These stages are executed using five modules, as depicted in Fig. 2: (1) exemplar extraction, (2) sliding window feature embedding, (3) exemplar-based similarity estimation, (4) exemplar-infused feature embedding, and (5) density estimation. In this section, we will elucidate these five modules along with the training procedure.

Exemplar Extraction

To extract the exemplars for the action class of interest, we first identify three temporal positions corresponding to the predefined vocal sounds (one, two, three) in the audio. A naive approach is to use a pre-trained classifier to greedily select the window with the highest classification score. However, this fails to exploit two critical cues: (1) temporal ordering, which requires the order of the sounds one, two, threes to be preserved, and (2) temporal proximity, which ensures that the distance between two predefined sounds is not excessively large. Considering these two properties, we formulate the temporal position detection into a constrained optimization problem as follows:

$$i^*, j^*, k^* = \underset{i,j,k}{\operatorname{argmax}}\, C_i^1 C_j^2 C_k^3, \tag{1}$$

$$\text{s.t. } 1 \le i < j < k \le M \text{ and } k - i \le R. \tag{2}$$

Here, i, j, k denote the indices of a sliding window. C_i^u is the classification score for the i^{th} window to be the u^{th} utterance. R is the upper bound for the temporal distance.

The above optimization problem can be solved efficiently using dynamic programming. We first divide the audio signal into M overlapping sliding windows, each with a duration of one second and the step size being 0.1 seconds. We then compute the classification scores (C_i^1, C_i^2, C_i^3) for each window using a pre-trained classifier, specifically the BC_ResNet (Kim et al. 2021) pretrained on Speech Command (Warden 2018). For every group of R consecutive windows, we optimize $C_i^1 C_j^2 C_k^3$ subject to the only constraint $i < j < k$ with dynamic programming. The complexity of this algorithm is $\mathcal{O}(R)$, and we have to run it $M - R + 1$ times for $M - R + 1$ groups of R consecutive windows. Thus, the overall complexity is $\mathcal{O}(R(M - R + 1))$.

Let $\mathcal{X} \in \mathbb{R}^{N\times d}$ denote the sensor data sequence, with N being the length and d the number of sensor values at each time step ($d = 6$ for data from the accelerometer and gyroscope of a smartwatch). Upon solving the above optimization problem, we obtain i^*, j^*, k^*, which indicate the locations of the three exemplars. To avoid noisy exemplars, we only retain the two locations with the highest classification confidence and let them be denoted as s_1 and s_2. Unfortunately, we do not know the temporal extents of the exemplars. To address this issue, we adopt a multi-scale approach as follows. For each position s among the two positions s_1 and s_2, we extract three exemplar sequences corresponding to three different scales: $\mathcal{X}[s-10 : s+10]$, $\mathcal{X}[s-20 : s+20]$, and $\mathcal{X}[s-40 : s+40]$. With two locations and three scales, we have a total of six exemplars. This strategy enables us to count actions at various levels of granularity.

Sliding Window Feature Embedding

As sensor values at individual time steps carry limited information, we learn and use window-level sensor representation instead. To accomplish this, we partition a sensor data sequence into non-overlapping windows, with each window

comprising w sensor data points. We subsequently embed each window into a high-dimensional representation turning the sequence of original sensor values $\mathcal{X} \in \mathbb{R}^{\frac{N}{w} \times d}$ into a sequence of embedding vectors $\mathcal{X}' \in \mathbb{R}^{\frac{N}{w} \times d'}$. Let ϕ denote this mapping, i.e., $\mathcal{X}' = \phi(\mathcal{X})$, and ϕ is implemented using temporal convolution. Specifically in our experiments, w is set to 10, and d' is set to 64. Similarly, the exemplar sequence $\mathcal{E}$ is transformed into $\mathcal{E}'$ using ϕ.

Exemplar-Based Similarity Estimation

Utilizing per-window embedding, we estimate the similarity map $\mathcal{S}$ between the sensor embedding $\mathcal{X}'$ and the exemplar embedding $\mathcal{E}'$. Correlation and Dynamic Time Warping (DTW) are two widely-used methods for estimating similarity between sequential data. However, directly applying them to estimate the similarity between $\mathcal{X}'$ and $\mathcal{E}'$ is not effective because correlation is sensitive to differences in scale and offset while DTW tends to overreact to static data. To address these issues, we combine DTW and correlation to estimate the similarity as follows.

We first compute the correlation between the whole sequence embedding and the exemplar embedding: $\mathcal{S}^c = ReLU(\text{Norm}(\mathcal{X}' \otimes \mathcal{E}'))$ where $\otimes$ is correlation operation with zero-padding to preserve the length of the signal (i.e., $\mathcal{S}^c$ and $\mathcal{X}'$ have the same length). Next, we calculate the Soft-DTW similarity (Cuturi and Blondel 2017) between the exemplar embedding and the sliding window on the whole sequence embedding. For the sliding window at location i, the resulting value is $\mathcal{S}^d_i = \text{Soft-DTW}(\mathcal{X}'[i - \frac{k}{2}, i + \frac{k}{2}], \mathcal{E}')$, where k is the length of the exemplar $\mathcal{E}'$. Then, $\mathcal{S}^d$ is fed into normalization and ReLU layers as $\mathcal{S}^d = ReLU(\text{Norm}(\text{Max}(\mathcal{S}^d) - \mathcal{S}^d))$. Considering that Soft-DTW estimates the distance between two samples, we transform it into a measure of similarity by taking the negative of the distance and adding the maximum value, thereby ensuring a non-negative similarity measure. The final similarity profile is obtained by computing $\mathcal{S} = \mathcal{S}^c \odot \mathcal{S}^d$, where $\odot$ denotes element-wise multiplication. Since we have two exemplars at three scales, the dimension of $\mathcal{S}$ is $\mathcal{S} \in \mathbb{R}^{\frac{N}{w} \times 6}$.

Exemplar-Infused Feature Embedding

Upon obtaining the similarity map $\mathcal{S}$, we use it to generate a refined representation that emphasizes exemplar-related features while suppressing irrelevant features. This can be implemented with a stack of K fusion blocks, and the process can be described as follows:

$$\mathcal{F}_0 = \mathcal{X}',\ \mathcal{S}_0 = \mathcal{S}, \tag{3}$$

$$\mathcal{S}_i = \text{CGAP}(\text{Conv}(\mathcal{S}_{i-1})), \tag{4}$$

$$\mathcal{F}_i = \text{Conv}(\mathcal{F}_{i-1} + \text{GELU}(\text{Norm}(\text{Conv}(\mathcal{F}_{i-1} \odot \mathcal{S}_{i-1})))).$$

Here, CGAP is the channel-wise (among exemplars) global average pooling, and $\odot$ denotes element-wise multiplication. The final fused feature is $\mathcal{F} = \mathcal{F}_K \in \mathbb{R}^{\frac{N}{w} \times d'}$.

Density Estimation

The density estimation head comprises a Feature Pyramid Network (FPN) designed to extract multi-scale features and a temporal convolution counting head ψ to estimate the temporal densities. We extract multi-scale features as follows:

$$\mathcal{F}_{s_1}, \mathcal{F}_{s_2}, \mathcal{F}_{s_3} = \text{FPN}(\mathcal{F}), \tag{5}$$

$$\mathcal{X}'_{s_1}, \mathcal{X}'_{s_2}, \mathcal{X}'_{s_3} = \text{FPN}(\text{Conv}(\mathcal{X}')), \tag{6}$$

where $\mathcal{F}_{s_1}, \mathcal{F}_{s_2}, \mathcal{F}_{s_3}$ are multi-scale fused features from low to high, and $\mathcal{X}'_{s_1}, \mathcal{X}'_{s_2}, \mathcal{X}'_{s_3}$ are multi-scale sensor feature for the sensor embedding. Using max-pooling, $\mathcal{F}_{s_1}, \mathcal{F}_{s_2}, \mathcal{X}'_{s_1}, \mathcal{X}'_{s_2}$ are down-sampled to have the same length as $\mathcal{F}_{s_3}$ and $\mathcal{X}'_{s_3}$. All of them are then concatenated and fed into a density estimation head ψ, implemented with a temporal convolution network.

Training Loss

The counting loss over the predicted temporal density map is given by the squared error of the final count, expressed as: $\mathfrak{L}_c = (\text{sum}(\mathcal{T}) - \hat{c})^2$, where $\hat{c}$ is the ground truth count.

The success of our method largely depends on accurately estimating similarity between the exemplars and the query data sequence. However, it's important to note that the similarity relationship within the raw data space $\mathcal{X}$ may not be fully preserved in the embedding space $\mathcal{X}'$. This is especially true when dealing with limited training data and the lack of a robust pre-trained feature extractor. Inspired by Laplacian Eigenmaps (Belkin and Niyogi 2003), we propose to use a distance-preserving loss to encourage the per-window encoder to preserve the relationship of distance by enforcing the encoder to maintain the local patterns. We first build a k-nearest-neighbor graph over the raw window to represent the local pattern. To build it, we compute the adjacency matrix $\mathcal{W}$, where $\mathcal{W}_{ij} = \exp(-\frac{||\mathcal{X}_i - \mathcal{X}_j||^2}{2\sigma^2})$ represents the similarity between the i^{th} window and j^{th} window. Then, for each node in the graph, we retain the top k nearest neighbors in the adjacency matrix ($k = 150$ in our work). We compute the graph Laplacian: $\mathcal{L} = \mathcal{D} - \mathcal{W}$, where $\mathcal{D}$ is the degree matrix with $\mathcal{D}_{ii} = \sum_j \mathcal{W}_{ij}$ and $\mathcal{D}_{ij} = 0$ for $i \neq j$. Then the distance-preserving loss is defined as $\mathfrak{L}_{pl} = \mathcal{X}'^T \mathcal{L} \mathcal{X}'$. The overall training loss is: $\mathfrak{L}_{train} = \mathfrak{L}_c + \lambda \mathfrak{L}_{pl}$, where λ is set to 0.01.

Pretraining with Synthesis Data

Given the difficulty of collecting data from wearable devices, the amount of training data will always be limited, and it is possible that the model may overfit to the training set and subsequently underperform when faced with out-of-distribution samples. To address the issue of dataset scarcity, we propose a data synthesis method. This approach leverages the predefined vocal sounds we previously discussed in the exemplar extraction section, effectively augmenting our existing dataset to bolster the model's robustness and ability to generalize. Our data synthesis approach consists of two main steps. Firstly, we mine action templates from an existing training set. Secondly, we randomly select a template and construct a sequence by aggregating multiple, randomly augmented versions of this template, interspersed with noise or repetitive irrelevant actions.

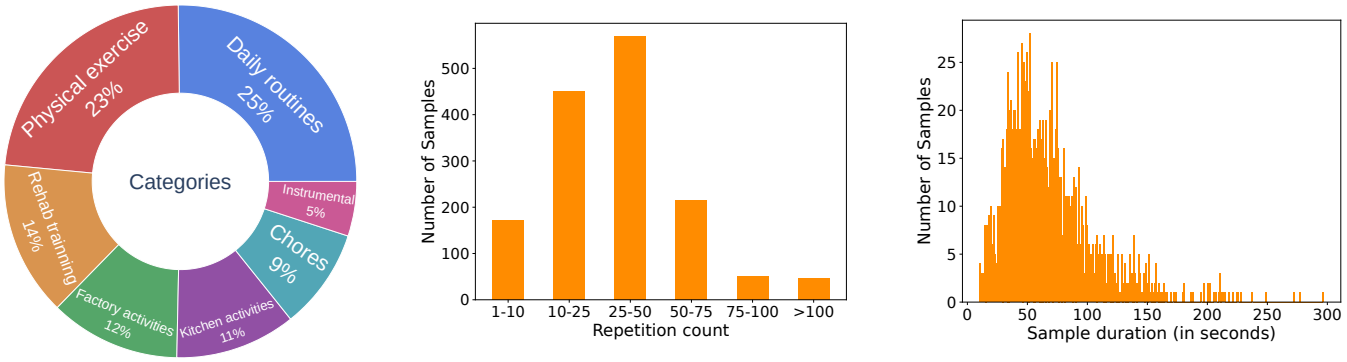


Figure 3: DWC dataset's statistics: The left figure displays the categories and the proportion of samples for each category in DWC. The two rightmost figures show the number of samples in various ranges of repetition count and duration.

Action template mining. In the exemplar extraction, we obtain the temporal positions i^*, j^*, k^* of the predefined utterances in the feature embedding sequence. We then remap these indexes to the time indexes $\hat{i}, \hat{j}, \hat{k}$ of the original sensor data sequence. Different from that, we retain the position with the minimum classification score during data synthesis. We consider $\mathcal{X}[\hat{i}{:}\hat{j}]$ and $\mathcal{X}[\hat{j}{:}\hat{k}]$ as two template candidates. We retain a candidate if it satisfies the following criteria: (1) Strong Confidence: the classification score of the temporal position greater than 0.75. This threshold ensures that we only select templates with a high degree of certainty, thus avoiding ambiguous cases. (2) Moderate Length: we discard template candidates that fall outside the established length bounds, thus avoiding excessively short or long templates that may not represent typical actions. By iterating through all the samples in the original training data, we construct an action template database, which serves as a foundation for synthesizing additional training data.

Action sequence generation with template. To synthesize a training sample, we first randomly sample one action template. Then we sample the count uniformly in the range $[0.8C_l, 1.2C_u]$, where C_l and C_u represent the minimum and maximum counts within the training set, respectively. Afterward, we aggregate c templates, augmenting each one through the following procedures: (1) duration scaling: we stretch or compress the duration of the template with scaling factor between 0.75 and 1.33; (2) time shifting: we shift the temporal position of the stretched/compressed template by random value within between -10 and 10 time steps; (3) Amplitude scaling: we modify the amplitude of the template by a scaling factor randomly chosen between 0.75 and 1.33; and (4) random noise addition: we introduce Gaussian noise with a standard deviation randomly chosen from 0 to 0.2.

Through these procedures, we ensure that each synthesized training sample embodies a diversity of temporal characteristics and amplitude variations, thus enriching the synthesized training sample. Upon aggregating c templates, we incorporate one to two irrelevant action sequences (described earlier) or static noise into the training sample. This integration is performed to mimic real-world data conditions, ensuring that our synthesized training data encapsulates a range of possible scenarios.

The DWC Dataset

Existing datasets for action counting from wearable devices (Mortazavi et al. 2014; Nishino, Maekawa, and Hara 2022; Zelman et al. 2020; Soro et al. 2019b; Prabhu, O'Connor, and Moran 2020; Strömbäck, Huang, and Radu 2020) often lack diversity in terms of both count values and action categories. Additionally, each data sample from these datasets also lacks diversity in terms of the actions contained within the sample, with the actions of interest being the predominant class. Considering these limitations, we introduce a more diverse dataset named DWC, which stands for Diverse and Wearable Counting. This dataset comprises 1502 entries of wearable-device data from 37 subjects across seven broad categories: kitchen activities, household chores, physical exercises, factory activities, daily routines, instrument-involved activities, and rehabilitation training. These broad categories encompass 50 distinct action classes, offering higher diversity compared to existing datasets.

We used a Samsung Galaxy Watch 4 for data collection. The sampling frequency was 100 Hz for both the 3-axis accelerometer and the 3-axis gyroscope, while the audio frequency was 16KHz. A total of 37 subjects were asked to wear the watch on their preferred hand while performing activities. Subjects were provided with a list of activities to perform in their chosen order. Each activity was accompanied by an illustrative guide and a brief textual description. The subjects were instructed to sequentially utter the words "one," "two," "three" while executing the first three repetitions of the action, with each utterance corresponding to one repetition. During data collection, participants could perform other types of action or take intermittent breaks. We manually inspected the collected data and annotated each sample with the number of repetitions of the action of interest. We also discarded samples in which the sensor and audio signals were not synchronized within 30ms. We developed an Android application to initiate the recording of both processes simultaneously, but since the audio stream was controlled by a third-party program, there were still instances of temporal mismatch.

The data was collected in two phases. In the first phase, 31 subjects participated, and each subject was asked to perform each of the 50 actions once. However, some subjects were

Method	Val Set		Test Set	
	MAE	RMSE	MAE	RMSE
Mean	17.18	21.91	14.80	17.49
Frequency-based	28.10	45.31	28.65	45.39
RepNet	11.95	17.33	10.82	14.75
TransRAC	14.51	20.40	12.97	16.82
Proposed	**7.66**	**12.25**	**7.47**	**13.09**

Table 1: Experiment results on DWC. The proposed method achieves the lowest counting errors. Note that the Test Set is completely disjoint from the Training Set.

Components	Combinations				
Pretrain	✗	✗	✗	✗	✓
Dist. Preserving Loss	✗	✗	✗	✓	✓
Constrained Detection	✗	✗	✓	✓	✓
Similarity Estimation	✗	✓	✓	✓	✓
MAE	11.30	10.87	10.32	10.05	**7.66**
RMSE	16.15	15.23	14.96	14.72	**12.25**

Table 2: Contributions of individual components

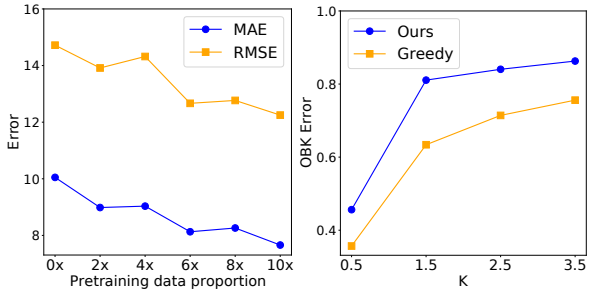


Figure 4: Left: model's performance as the amount of pretraining data is increased; "2x" represents twice the size of the real training set. Right: Quantitative result on temporal location detection. Off-By-K error under varying K.

not able to perform certain actions, such as push-ups, sit-ups, or jumping rope. The data collected in this phase containing 1356 entries with the action of interest occupying from 50% to 90% of the temporal duration. Upon completing the first phase, we recognized that the collected data did not possess sufficient diversity to address various practical scenarios that require counting non-dominant actions. Consequently, we proceeded with a second phase involving six additional subjects. We reviewed the list of 50 actions from the first phase and identified action classes that may not represent the predominant actions in realistic situations. Specifically, we selected six actions: picking up, shaking the clothes, slicing, tennis racket swinging, drinking and eating, and stretching. Each subject in the second phase was requested to perform each activity five times, although in some cases it was not feasible due to the lack of appropriate equipment. The data collected during this phase consists of 146 entries. These entries encompass more challenging samples where the action of interest constitutes a significantly smaller proportion of the temporal duration, ranging from only 10% to 20%. The final DWC dataset consists of 1502 entries, totaling 49,258 repetitions, and the statistics are shown in Fig. 3.

Experiments

Train, validation, and test data. We conducted experiments on the DWC dataset, using a partitioning scheme that guarantees the absence of shared subjects or action categories between the training and testing data. We first divided the data into two parts, containing 35 and 15 action categories, respectively. Within each part, we further separated the subjects into two groups, one containing 25 subjects and the other 12. The combination of the 35 action categories with 25 subjects became the training set, the 15 action categories with 12 subjects formed the test set, and the remaining data constituted the validation set.

Baselines. We compared the proposed method against four baseline models. *Mean* was a method that always outputted the mean count of the samples in the training data. *Frequency-based* was a method that predicted the final count based on the estimated the dominant frequency. We also compared with two state-of-the-art repetitive action counting methods, namely *RepNet* (Dwibedi et al. 2020) and *TransRAC* (Hu et al. 2022). To adapt these two methods for sensor data, we employed state-of-the-art feature extractors (Wu et al. 2021; Zhou et al. 2021; Liu et al. 2021) that were based on time-series forecasting and transfomers.

Evaluation metrics. Following almost all previous counting methods (e.g., Hu et al. (2022); Zhang, Shao, and Snoek (2021); Levy and Wolf (2015); Zhang et al. (2020); Zhang, Shao, and Snoek (2021)), we used Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as performance metrics, which are defined as: MAE $= \frac{1}{n}\sum_{i=1}^{n}|c_i - \hat{c}_i|$; RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(c_i - \hat{c}_i)^2}$, where n is the number of test samples, and c_i and $\hat{c}_i$ are the predicted and ground truth counts.

Implementation details. The training of our model proceeded in two stages. In the first stage, the model was pretrained on the synthesized data, which was ten times the volume of the actual training set, for 30 epochs using $\mathfrak{L}_{train}$ as the loss function. We utilized the Adam optimizer with a learning rate of 10^{-4} and a batch size of one for this pretraining. After pre-training, the model was trained on the actual training set for 30 epochs, using the same loss function, optimizer, and learning rate. The learning rate decay of 0.95 was applied at the end of each epoch.

During these two stages, the audio window classifier used in the exemplar extraction module was BC_ResNet (Kim et al. 2021), which was trained on Speech Command (Warden 2018) data. The classifier was frozen and not updated during the training stages. In our model, all input sensor data was padded to a common length of 28,000. For the baseline models, the feature extraction process involved embedding the sensor data into per-window embeddings, which were then fed into the feature extractor. We standardized the window size to 50 for all baseline feature extractors. Each feature extractor consisted of three layers with a specified hid-

den dimension of 256 and 8 attention heads. After feature extraction, the sensor features were passed through an adaptive pooling layer of size 96 before entering the counting head. The resulting temporal self-similarity map estimated by the counting head was then processed by an MLP to generate the temporal density map.

For RepNet, the input sensor data was padded to have the length of 28,000. TransRAC did not require padding. All models underwent a training phase of 60 epochs using the Adam optimizer with a learning rate of 10^{-5}. The training process was conducted with a batch size of one, and the count loss ($\mathfrak{L}_c$) was used as the loss function. All experiments were run on an RTX A5000 machine.

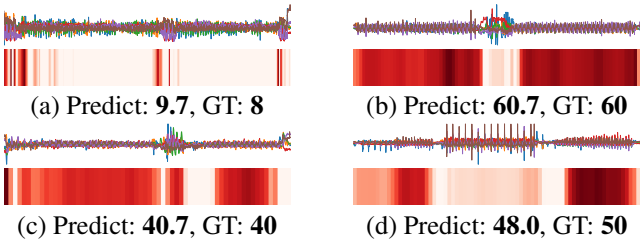
Quantitative results. Table 1 shows a performance comparison of various methods on the DWC dataset. The findings highlight the superiority of the proposed method, consistently achieving a minimum 30% lower MAE compared to other approaches. Notably, RepNet and TransRAC are strong baselines. For these baselines, extensive efforts were dedicated to optimizing their performance, tuning the pivotal feature extraction component of the methods, predominantly the time-series forecasting combined with a transformer architecture. In this pursuit, we explored a range of transformer variants, including the original transformer, Autoformer (Wu et al. 2021), Informer (Zhou et al. 2021), and Pyraformer (Liu et al. 2021). Specifically, the MAE values for RepNet on the test set, when using these transformer variants, are as follows: 10.82, 13.76, 11.99, 11.29, respectively. Likewise, the corresponding MAE values for TransRAC with these transformer variants are: 12.97, 14.12, 11.55, 12.99. Despite extensive efforts to tune their performance, the resulting MAE values for these methods remain at least 30% higher than our proposed method's MAE.

Ablation studies. To assess the effectiveness of each component in our proposed method, we conducted an ablation study using the validation data. The results of this analysis are presented in Table 2. The evaluated components include: (1) *Pretraining*: Referring to pretraining on the synthesized dataset; (2) *Dist. Preserving Loss*: Indicating the utilization of our distance-preserving loss; (3) *Constrained Detection*: Representing the use of our dynamic programming algorithm to detect the temporal locations of counting utterances under the temporal ordering and temporal proximity constraints. In its absence, we would employ a naive solution that selects the audio window with the highest classification score; and (4) *Similarity Estimation*: Indicating the proposed method for exemplar similarity estimation. In its absence, we use a naive correlation to estimate the similarity. The results presented in Table 2 demonstrate the beneficial impact of all proposed components on the overall performance. Particularly noteworthy is the significant contribution of pretraining on the synthesized dataset, which had the most substantial effect on the final result.

Given that pretraining is the most crucial component, we conducted further analysis to examine the impact of different amounts of pretraining data. In our default setting, we adopted an aggressive strategy, incorporating a large volume of synthesized training data, which is ten times the size of the real training data. However, we wanted to investigate whether a smaller amount of synthesized data could still yield significant improvements, resulting in faster pretraining. The results of this experiment are shown in Fig. 4(a), where different proportions of the default synthesized data were used (with random selection). Specifically, "2x" represents twice the size of the real training set, and "4x" indicates four times the size. Intriguingly, our results reveal that even a synthesized dataset only twice the size of the real training data leads to a marked improvement in performance. Additionally, we assessed the effectiveness of using a different number of exemplars, as presented in Table 3.

One exemplar		Two exemplars		Three exemplars	
MAE	RMSE	MAE	RMSE	MAE	RMSE
9.08	14.88	8.74	14.29	7.66	12.25

Table 3: Experiment results on the proposed DWC validation set with different numbers of audio exemplars.



(a) Predict: **9.7**, GT: **8**

(b) Predict: **60.7**, GT: **60**

(c) Predict: **40.7**, GT: **40**

(d) Predict: **48.0**, GT: **50**

Figure 5: Qualitative results. Four prediction examples. Each example shows the input sensor data, the estimated density map, the predicted count, and the ground truth value.

Quantitative analysis for exemplar localization. Our approach relies heavily on the temporal localization of the predefined utterances. To evaluate its efficacy, we conducted an experiment on the validation set, and the result is shown in Fig. 4. For evaluation, we used the Off-By-K Error (OBK) metric, defined as: $\text{OBK} = \frac{1}{N}\sum_{i=1}^{N}\delta(|t_i - \hat{t}_i| \leq K)$. Here, δ is the Diract delta function, N represents the total number of temporal locations, t_i is the predicted temporal location, and $\hat{t}_i$ is the ground truth temporal location. This metric measures the temporal discrepancy in seconds, between a predicted location and its corresponding ground truth location. We set a naive greedy scheme as the baseline for comparison. The results of our experiment underscored the effectiveness of our approach.

Qualitative results. Qualitative results shown in Fig. 5 demonstrate our method's ability to accurately leverage the exemplars for counting the actions of interest.

Conclusions

We propose a few-shot method for counting actions in real-world settings, utilizing vocal sounds from audio data to gather exemplars. These exemplars efficiently estimate action frequency over time. Our approach, validated by a comprehensive dataset, has proven effective in evaluations.

Acknowledgements

This project was partially supported by US National Science Foundation Award NSDF DUE-2055406 and AFOSR Award FA2386-23-1-4058.

References

Azy, O.; and Ahuja, N. 2008. Segmentation of periodically moving objects. In *Proceedings of the International Conference on Pattern Recognition*.

Baghdadi, A.; Cavuoto, L. A.; Jones-Farmer, A.; Rigdon, S. E.; Esfahani, E. T.; and Megahed, F. M. 2021. Monitoring worker fatigue using wearable devices: A case study to detect changes in gait parameters. *Journal of quality technology*, 53(1): 47–71.

Belkin, M.; and Niyogi, P. 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.*, 1373–1396.

Bian, S.; Rey, V. F.; Hevesi, P.; and Lukowicz, P. 2019. Passive capacitive based approach for full body gym workout recognition and counting. In *Proceedings of the International Conference on Pervasive Computing and Communications*.

Chang, K.-h.; Chen, M. Y.; and Canny, J. 2007. Tracking free-weight exercises. In *Proceedings of the ACM international joint conference on Pervasive and Ubiquitous Computing*.

Chetverikov, D.; and Fazekas, S. 2006. On Motion Periodicity of Dynamic Textures. In *Proceedings of the British Machine Vision Conference*.

Cutler, R.; and Davis, L. S. 2000. Robust Real-Time Periodic Motion Detection, Analysis, and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8): 781–796.

Cuturi, M.; and Blondel, M. 2017. Soft-DTW: a Differentiable Loss Function for Time-Series. In *Proceedings of the International Conference on Machine Learning*.

Ding, H.; Shangguan, L.; Yang, Z.; Han, J.; Zhou, Z.; Yang, P.; Xi, W.; and Zhao, J. 2015. Femo: A platform for free-weight exercise monitoring with rfids. In *Proceedings of the ACM conference on embedded networked sensor systems*.

Dwibedi, D.; Aytar, Y.; Tompson, J.; Sermanet, P.; and Zisserman, A. 2020. Counting Out Time: Class Agnostic Video Repetition Counting in the Wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Fieraru, M.; Zanfir, M.; Pirlea, S. C.; Olaru, V.; and Sminchisescu, C. 2021. AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Genovese, V.; Mannini, A.; and Sabatini, A. M. 2017. A smartwatch step counter for slow and intermittent ambulation. *Ieee Access*, 5: 13028–13037.

Hatamie, A.; Angizi, S.; Kumar, S.; Pandey, C. M.; Simchi, A.; Willander, M.; and Malhotra, B. D. 2020. Textile based chemical and physical sensors for healthcare monitoring. *Journal of the electrochemical society*, 167(3): 037546.

Hsu, Y.; Zhang, Q.; Tsougenis, E.; and Tsui, K. 2021. Viewpoint-Invariant Exercise Repetition Counting. *CoRR*.

Hu, H.; Dong, S.; Zhao, Y.; Lian, D.; Li, Z.; and Gao, S. 2022. TransRAC: Encoding Multi-scale Temporal Correlation with Transformers for Repetitive Action Counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Huang, Y.; Ranjan, V.; and Hoai, M. 2023. Interactive Class-Agnostic Object Counting. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Ishii, S.; Nkurikiyeyezu, K.; Luimula, M.; Yokokubo, A.; and Lopez, G. 2021. Exersense: real-time physical exercise segmentation, classification, and counting algorithm using an imu sensor. *Activity and Behavior Computing*, 239–255.

Kim, B.; Chang, S.; Lee, J.; and Sung, D. 2021. Broadcasted Residual Learning for Efficient Keyword Spotting. In *Proceedings of the Annual Conference of the International Speech Communication Association*.

Kong, X. T.; Luo, H.; Huang, G. Q.; and Yang, X. 2019. Industrial wearable system: the human-centric empowering technology in Industry 4.0. *Journal of Intelligent Manufacturing*, 30: 2853–2869.

Kranz, M.; Möller, A.; Hammerla, N.; Diewald, S.; Plötz, T.; Olivier, P.; and Roalter, L. 2013. The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices. *Pervasive and Mobile Computing*, 9(2): 203–215.

Kupke, J.; Willemsen, T.; Keller, F.; and Sternberg, H. 2016. Development of a step counter based on artificial neural networks. *Journal of Location Based Services*, 10(3): 161–177.

Lee, H. J.; Hwang, S. H.; Yoon, H. N.; Lee, W. K.; and Park, K. S. 2015. Heart rate variability monitoring during sleep based on capacitively coupled textile electrodes on a bed. *Sensors*, 15(5): 11295–11311.

Levy, O.; and Wolf, L. 2015. Live Repetition Counting. In *Proceedings of the IEEE International Conference on Computer Vision*.

Liu, C.; Zhong, Y.; Zisserman, A.; and Xie, W. 2022. CounTR: Transformer-based Generalised Visual Counting. In *Proceedings of the British Machine Vision Conference*.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *Proceedings of the International conference on learning representations*.

Lu, E.; Xie, W.; and Zisserman, A. 2018. Class-Agnostic Counting. In *Proceedings of the Asian Conference on Computer Vision*.

Morris, D.; Saponas, T. S.; Guillory, A.; and Kelner, I. 2014. RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Mortazavi, B. J.; Pourhomayoun, M.; Alsheikh, G.; Alshurafa, N.; Lee, S. I.; and Sarrafzadeh, M. 2014. Determining the Single Best Axis for Exercise Repetition Recognition

and Counting on SmartWatches. In *Proceedings of the International Conference on Wearable and Implantable Body Sensor Networks*.

Nam, Y.; Kim, Y.; and Lee, J. 2016. Sleep monitoring based on a tri-axial accelerometer and a pressure sensor. *Sensors*, 16(5): 750.

Nguyen, T.; Pham, C.; Nguyen, K.; and Hoai, M. 2022. Few-Shot Object Counting and Detection. In *Proceedings of the European Conference on Computer Vision*.

Nishino, Y.; Maekawa, T.; and Hara, T. 2022. Few-Shot and Weakly Supervised Repetition Counting With Body-Worn Accelerometers. In *Frontiers in Computer Science*.

Oh, M.-h.; Olsen, P.; and Ramamurthy, K. N. 2020. Crowd counting with decomposed uncertainty. In *Proceedings of the AAAI conference on artificial intelligence*.

O'Reilly, M.; Caulfield, B.; Ward, T.; Johnston, W.; and Doherty, C. 2018. Wearable inertial sensor systems for lower limb exercise detection and evaluation: a systematic review. *Sports Medicine*, 48: 1221–1246.

Patel, S.; Hughes, R.; Hester, T.; Stein, J.; Akay, M.; Dy, J. G.; and Bonato, P. 2010. A novel approach to monitor rehabilitation outcomes in stroke survivors using wearable technology. *Proceedings of the IEEE*, 98(3): 450–461.

Pillai, A.; Lea, H.; Khan, F.; and Dennis, G. 2020. Personalized step counting using wearable sensors: A domain adapted LSTM network approach. *arXiv preprint arXiv:2012.08975*.

Pogalin, E.; Smeulders, A. W. M.; and Thean, A. H. C. 2008. Visual quasi-periodicity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Prabhu, G.; O'Connor, N. E.; and Moran, K. 2020. Recognition and Repetition Counting for Local Muscular Endurance Exercises in Exercise-Based Rehabilitation: A Comparative Study Using Artificial Intelligence Models. *Sensors*, 20.

Ramachandran, B.; and Liao, Y.-C. 2022. Microfluidic wearable electrochemical sweat sensors for health monitoring. *Biomicrofluidics*, 16(5): 051501.

Ranjan, V.; and Hoai, M. 2022a. Exemplar Free Class Agnostic Counting. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

Ranjan, V.; and Hoai, M. 2022b. Vicinal Counting Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Ranjan, V.; Sharma, U.; Nguyen, T.; and Hoai, M. 2021. Learning To Count Everything. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Runia, T. F. H.; Snoek, C. G. M.; and Smeulders, A. W. M. 2018. Real-World Repetition Estimation by Div, Grad and Curl. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Shi, M.; Lu, H.; Feng, C.; Liu, C.; and Cao, Z. 2022. Represent, Compare, and Learn: A Similarity-Aware Framework for Class-Agnostic Counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Soro, A.; Brunner, G.; Tanner, S.; and Wattenhofer, R. 2019a. Recognition and repetition counting for complex physical exercises with deep learning. *Sensors*, 19(3): 714.

Soro, A.; Brunner, G.; Tanner, S.; and Wattenhofer, R. 2019b. Recognition and Repetition Counting for Complex Physical Exercises with Deep Learning. *Sensors*, 714.

Stiefmeier, T.; Roggen, D.; Ogris, G.; Lukowicz, P.; and Tröster, G. 2008. Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing*, 7(2): 42–50.

Strömbäck, D.; Huang, S.; and Radu, V. 2020. MM-Fit: Multimodal Deep Learning for Automatic Exercise Logging across Sensing Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4.

Thangali, A.; and Sclaroff, S. 2005. Periodic Motion Detection and Estimation via Space-Time Sampling. In *Proceedings of the Applications of Computer Vision Workshop*.

Warden, P. 2018. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *CoRR*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*.

Yang, S.; Su, H.; Hsu, W. H.; and Chen, W. 2021. Class-agnostic Few-shot Object Counting. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*.

You, Z.; Yang, K.; Luo, W.; Lu, X.; Cui, L.; and Le, X. 2023. Few-shot Object Counting with Similarity-Aware Feature Enhancement. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*.

Zelman, S.; Dow, M. M.; Tabashum, T.; Xiao, T.; and Albert, M. V. 2020. Accelerometer-Based Automated Counting of Ten Exercises without Exercise-Specific Training or Tuning. *Journal of Healthcare Engineering*.

Zhang, H.; Xu, X.; Han, G.; and He, S. 2020. Context-Aware and Scale-Insensitive Temporal Repetition Counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, Y.; Shao, L.; and Snoek, C. G. M. 2021. Repetitive Activity Counting by Sight and Sound. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI Conference on Artificial Intelligence*.