Learning Global Context for Sparse Activity Recognition in Lengthy Recordings with Limited Dataset Size

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Computer Engineering

---

by
Zeyu Tang
May 2026

Accepted by:
Dr. Adam Hoover, Committee Chair
Dr. Jon Calhoun
Dr. Feng Luo
Dr. Lan Zhang

Abstract

This dissertation describes methods to analyze lengthy recordings of data in order to detect sparsely occurring activities. The narrative below describes the progression of research that led to the development of these methods and their generalization into a unified framework.

My research started with designing models for dietary monitoring, including detecting meals from day-long recordings and detecting intake gestures from meal-length recordings. Both tasks share some common characteristics: (a) the target event takes only a small portion of data recordings, and (b) there is global context within full-length data recordings that can help a model make better decisions. After finishing these projects and making several publications, we summarize a general problem definition for this group of tasks and a general methodological pipeline to effectively leverage global context within full-length data recordings to improve model performance.

We define events that are either temporally brief or infrequent in data as **sparse events**. Traditional methods for detecting sparse events are inherited from the general time-sequence analysis pipeline and rely on sliding-window classifiers. This pipeline slices recordings into short windows and classifies each window as an independent sample. Most existing works focus on extracting richer representations from each window, leading to a variety of models based on CNNs, RNNs, and transformers. However, through our exploration of detecting meals from day recordings and detecting eating gestures from meal recordings, we found that window-based classifiers struggle with false positives and limited context, particularly when only short segments are analyzed in isolation. These limitations stem from the scarcity of positive samples and temporal ambiguity between events and background activity. These natural weaknesses of window-based classifiers motivated our research to explore the potential of including broader recording-level context for event detection.

A few researchers have noted the advantages of jointly analyzing neighboring windows, but such approaches are constrained by increased computational demands and the need for larger

datasets. When treating multiple windows as a single sample, models must be scaled up to handle larger input sizes, and each recording yields fewer samples than in single-window approaches.

The proposed unified framework for sparse-event detection achieves global-pattern modeling on full-length recordings while maintaining efficiency in both data usage and computation. The core idea is to combine a local feature encoder, which compresses window-based data into smaller vectors, with a global detector, which captures long-range dependencies and recording-level contextual patterns. To address the challenge of limited data, we propose a novel augmentation method that generates synthetic global patterns and improves the generalization capacity of global detectors.

Chapters 1-3 incrementally introduce my work on developing a framework specifically for two tasks: detecting meals and detecting intake gestures. Each case begins with data preparation and observation, followed by methodological refinements tailored to the dataset characteristics. Chapter 1 presents a global-local detection model for eating episode detection from full-day wrist-motion data and shows that modeling daily patterns reduces false positives and improves generalizability. Chapter 2 introduces a video dataset tailored to sparse intake-gesture detection, which forms the foundation for intake-gesture recognition in free-living environments. Chapter 3 presents a global-local detection model applied to meal-length videos and demonstrates that leveraging full-meal context improves performance, particularly precision and training stability.

In Chapter 4, we evaluate our global pattern analysis framework under a more systematic and scalable structure. We apply a similar idea about global context to two additional case studies: sleep-stage detection from EEG signals and speech/music detection from TV-show audio. These studies expand the framework across modalities (vision, motion, biology, and audio), recording durations (from 30 minutes to 24 hours), and dataset sizes (from 300 to 2000 recordings). We summarize a standard framework, examine performance gains across different conditions, and explore both the potential and limitations of our framework.

Acknowledgments

I am deeply grateful to my supervisor, Dr. Adam Hoover, for his guidance, trust, and steadfast support throughout my graduate journey. His character and mentorship have shaped not only the rigor of my research, but also my attitude toward life and career.

I am profoundly thankful to my parents for their consistent support and patience. During periods of stress and uncertainty about the future, their love and encouragement have given me the strength and spirit to keep moving forward.

I thank my committee members, Dr. Jon Calhoun, Dr. Feng Luo, Dr. Lan Zhang, and Dr. Kuang-Ching Wang (who departed after my comprehensive exam), for their time and interest in my research.

I sincerely thank my lab colleague James Jolly, who has been a warm collaborator since I joined the lab in 2019. I am especially grateful to Sharma Surya for mentoring me and helping me ramp up when I first joined the lab. I also thank Adam Patyk, with whom I worked for about two years on my first funded project. I also acknowledge Yu Xuan, Brycen Havens, and Faria Armin, whose presence has contributed to the growth and prosperity of our lab.

I would also like to acknowledge the love and support of the greater Clemson family (faculty, staff, and friends), who have supported and witnessed my growth since my first days as a graduate student.

Finally, I sincerely acknowledge the following grants, which have supported my research and provided opportunities to tackle real-world projects: NIH R01DK135679, "Validating Sensor-based Approaches for Monitoring Eating Behavior and Energy Intake by Accounting for Real-World Factors that Impact Accuracy and Acceptability"; NIH R01DK132210, "Using Multimodal Real-Time Assessment to Phenotype Dietary Non-Adherence Behaviors that Contribute to Poor Outcomes in Behavioral Obesity Treatment"; NIH R01HL153543, "Optimizing Just-in-Time Adaptive Interven-

tion to Improve Dietary Adherence in Behavioral Obesity Treatment: A Micro-randomized Trial"; NSF 2242812, "AI-enabled Devices for the Advancement of Personalized and Transformative Healthcare in South Carolina"; DARPA Arcadia program, "Engineering Control of Organic Coatings on Autonomous Navy Gliders"; and Samsung and South Carolina Department of Commerce, "Advanced Visual Inspection System."

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, Sections 1.1 through 1.4 present the motivation and completed domain studies in dietary monitoring. We then connect these studies through a shared sparse-event-recognition perspective and summarize the key novelties of the dissertation in Section 1.5.

Section 1.6 introduces the generalized, domain-agnostic framework. It defines the core time-series setting, reviews standard sequence-modeling methodologies, and links the completed dietary-monitoring studies to two extension domains: sleep stage detection from EEG and speech/music detection from audio.

1.1 Motivation

My research began with designing models for dietary monitoring. I completed two modeling tasks in this domain: detecting eating episodes from wrist motion data [99], and detecting intake gestures from videos [98]. For each task, we developed a separate method to leverage long-range context within full-length recordings and improve the performance of existing models.

After completing the modeling work for dietary monitoring, we began to realize that the detection of eating episodes and intake gestures shares some common factors. First, the target events in both tasks represent minor classes within the recordings. Second, in the data recordings, whether day-long wrist motion data or meal-length videos, there are global patterns that span the timespan of each recording and remain consistent across individuals. Our experiments have shown that these global patterns can be learned by neural networks and can help improve detection

performance [98, 99].

We hypothesize that our methods for learning global context from full-length recordings can be unified into a general framework. Such a framework can be applied to additional application domains with similar problem characteristics. This has become the main focus of our current research.

The same problem structure appears beyond dietary monitoring. In lengthy recordings, target events may be infrequent (for example, meal episodes in day-long IMU data) or brief and intermittent (for example, speech/music segments in long-form audio and stage transitions in all-night EEG). In all cases, target events occupy a minority of the timeline, while long-range context across the full recording remains informative.

Most prior approaches in these domains use local window-based analysis [24, 26, 27, 49, 75, 82, 86, 95]. This strategy learns local patterns effectively, but it can underuse full-recording temporal structure. Moving to full-recording modeling introduces a second challenge: limited sample size, since one full recording typically contributes one training sample rather than many windows.

This setting is related to anomaly detection, but differs in an important way. Many anomaly-detection pipelines treat target events as outliers [60, 89], whereas the events studied in this dissertation are sparse but semantically structured and recurrent, enabling explicit supervised modeling. The core challenge is to learn global context from full-length recordings under limited data.

This perspective motivates the top-down two-stage framework used throughout this dissertation: first extract informative local representations, then model global temporal structure at the recording level. With this framing in place, we next describe the dietary-monitoring application context.

1.2 Dietary Monitoring System

Unhealthy dietary habits are a key behavioral risk factor associated with noncommunicable diseases (NCDs), which contribute to 70% of global deaths [66]. More than 1 billion people in the world are classified as obese, including more than 340 million adolescents and 39 million children under the age of 5 [37, 67, 109].

The need to address unhealthy dietary habits motivates the development of practical methods for monitoring daily food intake. Traditional tools for dietary monitoring, such as food diaries

and 24-hour recalls, impose a high user burden that lead to decreased compliance [79]. And their measurements are distorted by self-reporting bias and inaccuracy [9, 12]. Consequently, current research focuses on automating the monitoring of daily intake activities. These automated tools facilitate the objective quantification of eating behaviors, encompassing aspects such as meal duration, food and beverage consumption [38, 70], and the enumeration of intake occurrences [47, 75]. These metrics yield valuable insights and increase the potential for treatment options [2].

Encouraging recent research has shown that time-restricted eating, in which patients are encouraged to limit intake to a window of 8-12 hr per day, can significantly influence weight loss [28,32]. To support this type of treatment, new tools are needed which provide all-day measurements of eating behavior with minimal burden.

Researchers have investigated various sensor modalities for detecting dietary intake activities. Specifically, significant efforts have been dedicated to wearable devices equipped with sensors, including inertial measurement unit sensors [21, 22, 58], acoustic sensors [29, 68, 69] and electromyography sensors [48, 114, 115]. These wearable devices can be worn on the wrist or head, enabling users to carry them wherever they go. However, they require frequent recharging and may disrupt the daily routine of users due to the need for continuous wear.

A camera, in contrast to wearable sensors, offers an environment-based solution for detecting intake activities [47, 75]. Once installed, they can operate with no user burden, eliminating the need for daily charging and continuous wear, thus reducing disruptions to daily routines. Another notable advantage of camera sensors is their capacity to provide insights into the types of foods and beverages being consumed [38].

Both wearable devices and camera-based monitoring systems have advantages and disadvantages. An advantage of a wearable device is that it can be worn everywhere the user goes to monitor eating. A disadvantage is that it must be recharged daily, and the user must remember to put it on every day. An advantage of a stationary camera-based system is that after initial installation no further effort is required from the user, but a disadvantage is that the camera can only monitor eating at the instrumented location. Another advantage of camera systems is that they can provide information about the types of foods and beverages being consumed [38]. Researchers have also used cameras as retrospective memory aids to help users recall and record eating activities [30, 44, 93]. However, cameras must be used carefully to protect privacy. We believe both approaches will find practical applications as the need for automated dietary monitoring increases.

1.3 Domain Research: Detecting Eating Episodes From Wrist Motion Using Daily Pattern Analysis

Researchers are actively investigating new methods for automating the monitoring of eating using wearable devices [51]. Examples include wrist-worn devices that can measure hand-to-mouth gestures (moments of intake) [41], eyeglass and earpiece devices that can measure motions and sounds associated with mastication (chews) [81], and throat-located devices that can measure forces and sounds associated with ingestion (swallows) [107].

1.3.1 Related Work

Previous works attempt to detect individual ingestion or consumption events first. Periods of eating (i.e. meals, snacks) are then inferred by grouping individual events into spans of time. For example, Doulah et al. used accelerometers and flex sensors mounted on eyeglasses to detect individual intake events, and then smoothed these detections to identify the start and end boundaries of eating episodes (meals, snacks) [23]. Mirtchouk et al. used smartwatch IMU sensors and earbud-mounted microphones to detect chews and intake gestures, and then combined them across gaps of less than 1 minute to identify entire meal periods [62, 63]. Rouast et al. developed a neural network to recognize individual intake events in wrist motion data and in video data [76]. Kyritsis et al. developed a similar neural network to recognize individual intake events [55], and then developed a second-stage approach to group them together to detect entire meal periods [56].

The prior research on meal detection, as outlined above, employs a *bottom-up* methodology. This strategy initially identifies brief ingestion or consumption events within a small data window and then relies on the heuristic aggregation of these detection results over time to locate larger periods of eating. The challenge for these bottom-up approaches is that they are facing the "needle in a haystack" problem, since intake gestures and chewing sounds last only a few seconds each and are infrequent events in a full day. Additionally, other actions with similar motion patterns can occur which are unrelated to eating. This makes bottom-up approaches vulnerable to lots of false positives, as demonstrated in the experimental results presented in our research. It also misses the opportunity to utilize the context of when eating episodes (meals, snacks) are more likely to occur within a day-length period of time. The experiments conducted in our research illustrate that by considering these long-term contexts, both episode TPR and FP/TP of existing window-based

models in meal detection can be substantially improved.

1.3.2 Challenges and Contributions in this Research

The main contribution of our work is the discovery that day-length analysis of wrist motion data notably improves the detection of eating compared to shorter local window analysis. We describe a novel *top-down approach*, in which we analyze an entire day of data as a single sample. From that broad perspective we identify periods of time during which eating occurred (meals, snacks), without relying upon explicit detection of the individual ingestion or consumption events that occurred within those periods.

By analyzing an entire day of data as a single sample, the analysis can take advantage of diurnal context that can impact eating habits, such as the time of day, time since waking, and time since last meal, all of which could help improve the detection of when eating is occurring. It is important to note that diurnal context cannot be modeled by simple statistics or probabilities as previous research has found wide variability both between and within individuals comparing day-to-day patterns of eating [32]. Therefore we use a neural network classifier which is capable of learning complex patterns beyond simple probabilities based on time-of-day.

In order to make training networks on day-long samples computationally feasible, we implement a two-stage framework to limit the computational burden. A full day of wrist motion data contains appx 8 M values (e.g. 24 hr of gyroscope and accelerometer data recorded at 15 Hz). This would require tens of millions of parameters to analyze using an encoder-decoder model to analyze the entire day of data and label it all at once; e.g. the popular ResNet50 uses appx 23 M parameters to analyze an image containing 0.15 M values [40]. Instead, by breaking the analysis into two steps, we can redefine the first problem as targeting a single label for a longer window of time (e.g. 1-2 minutes) as "eating" or "not eating". Once the eating periods are identified for a day, the individual data within those periods can be reanalyzed at higher resolution using ingestion event labels (e.g. targets of "bite" or "drink" as in [58]). Using this approach, we show how to build a first-stage model for window-based classification and a second-stage model that needs only approximately 1 K parameters for detecting eating. This has practical implications for a model intended to run on wearable device hardware where computational power and battery life are limited.

In order to generate an adequate number of day-length samples for training a second-stage model, we apply a new data augmentation technique. Training a neural network model, even of

moderate size, typically requires a large number of training data (e.g. training ResNet50 typically requires over 1 M images from the ImageNet dataset [40]). For previous approaches that analyze a few seconds or minutes of data per sample, day-length recordings from 10-100 people can be used to generate 10,000+ window samples by intermittently cutting windows from the recordings [55, 76, 85]. However, we seek to analyze 24 hours of data per sample, so each day-length recording provides only one sample. To overcome this challenge we propose a novel data augmentation technique that leverages model volatility. Repeated retraining of our stage 1 model produces variations in the instantaneous probability $P(E_w)$. We use this model volatility to augment our dataset by over 500x, to train and test our stage 2 model.

We evaluate the effect of day-length analysis on two state-of-the-art window-based eating detectors [42, 85] on two publicly available datasets: Clemson All-day (CAD)[1] and FreeFIC[2]. In all cases we demonstrate a substantial improvement in accuracy in the detection of periods of eating. Finally, we use the CAD dataset as a benchmark (it is currently the largest available dataset of day-length wrist motion recordings, containing 354 day-length recordings from 351 people [84]) and compare our new method against several previous works. We demonstrate that our method achieves the highest accuracy to date on this dataset.

The novelty of this research can be summarized as follows:

1. We present methods to analyze a full day of wrist motion data as a single sample so that the detection of eating occurrences can benefit from diurnal context.

2. We describe a two-stage framework to daily pattern analysis that calculates probabilities by examining local windows and subsequently extending this analysis to encompass entire days. In this framework, we demonstrate how many local eating detectors could be applied by showing two examples.

3. We demonstrate that false positive detections are substantially reduced by daily pattern analysis compared to local window analysis.

Detailed methodology, experiments and results are included in 2.

[1] `https://cecas.clemson.edu/ahoover/allday/`
[2] `https://zenodo.org/record/4421951`

1.4 Domain Research: Video-based Intake Gesture Recognition using Meal-length Context

Accurate detection of intake gestures, specifically bites and drinks, is crucial for dietary monitoring systems to reliably assess eating behaviors. These measurements directly facilitate evaluations such as eating rate (bites/min), total bite gesture count (a proxy for portion size), and total drink gesture count (used alongside bite count to calculate the liquid-to-solid intake ratio). Additionally, there is growing evidence that the total bite count may serve as a reasonable proxy for meal-level energy intake (kilocalories). For instance, Salley et al. demonstrated that bite-based estimates of kilocalories were more accurate than participants' guesses across 271 individuals each eating a single meal in a cafeteria setting [78]. Similarly, Chou et al. discovered that bite count contributed more significantly to total energy intake than the types of foods consumed [17]. These assessments of eating behaviors, distinguishing between lean and obese individuals, are instrumental in formulating recommendations for behavioral changes to manage obesity [90]. This research provides evidence that measurements of intake gestures can be effectively extracted from video data, supporting the development of home-based systems for dietary monitoring.

1.4.1 Related Work

1.4.1.1 Window-based Video Analysis

Some researchers have focused on developing deep learning models tailored for the assessment of eating behaviors through video analysis [47,75,81,104]. Rouast et al. explored various video models and identified SlowFast [27] and a custom-designed recurrent neural network (RNN), which they called CNN-LSTM, as the top-performing models for detecting intake gestures [75]. SlowFast employs two kernels to scan input data at different temporal speeds [27]. In CNN-LSTM, spatial-temporal information is encoded using a CNN, and the resulting feature maps are then aggregated via long short-term memory (LSTM) [43], for video-level predictions. In a subsequent study, Rouast et al. delved into the utilization of connectionist temporal classification (CTC) loss to train a single-stage model for detecting and localizing intake gestures [76]. This approach enabled the model to learn gesture sequencing instead of treating each gesture independently. However, both of these studies had a limitation in that they analyzed relatively short data windows, ranging from 2 seconds

to 8 seconds.

Hossain et al. developed methods to quantify both bites and chews from video recordings over a longer duration [47]. Their approach involved initially using Faster R-CNN to extract facial regions from videos. Subsequently, they applied AlexNet to classify each frame as either a bite or non-bite, followed by utilizing 1D optical flow to tally chews over a maximum time span of 52 seconds. However, it is important to mention that this final post-processing step was heuristic and not learned by any specific model.

We theorize that eating-related gestures inherently carry sequential context that can be harnessed to improve their recognition over extended time periods. For instance, food preparation activities such as cutting and stirring are typically followed by the intake of food, leading to mastication [73]. Bite gestures, indicative of food intake, tend to occur more frequently than drink gestures. Beverage intake often happens more toward the end of a meal, informally referred to as "washing down food", and intake tends to slow as the person eating becomes satiated. These contextual cues within a meal-length timeframe hold the potential for enhancing the recognition of intake gestures.

In the following section, we will review related work focused on the analysis of full-length videos to elucidate concepts explored in diverse applications unrelated to dietary intake.

1.4.1.2 Full-length Video Analysis

Several researchers have attempted to leverage long-term temporal information in videos within an affordable computational overhead. One approach involves splitting videos into segments and compressing each segment into a feature vector. For example, Tu et al. [103] adaptively split videos into segments and aggregated segment-level deep feature maps over the video to classify videos from a video-level representation. However, their feature encoding method is limited to video-wise classification and is not suitable in scenarios such as intake gesture detection, where frame-wise classification is required for temporally locating actions in videos. Additionally, their method was found to outperform window-based networks only on short-trimmed videos with actions of interest taking up most of the video [103]. Meal videos, on the other hand, are tens of minutes long and contain numerous non-intake activities.

Another research direction involves extracting long-term information and integrating it into window-based networks to classify short video clips. For example, Yang et al. [112] introduced a collaborative memory pool shared by every clip windowed from the same video. By simultane-

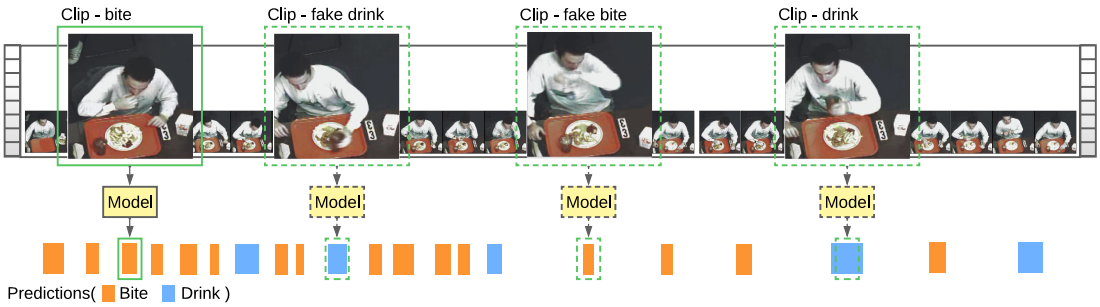
ously applying a window-based network to every clip within a video, the memory pool gathered features from each clip and infused these features into each clip. In another work, Tang et al. [96] introduced a global attention mechanism that employed long-term attention features to enhance window-based networks. However, neither the collaborative memory mechanism nor the global attention mechanism effectively addressed the computational challenges when processing long videos. For instance, Tang et al. had to limit the maximum input frame count to 768 during model testing on ActivityNet-1.3 due to GPU memory constraints.

All of these prior works either focus solely on clip-level interaction, neglecting long-term context at different scales, or concentrate on short videos with durations of several minutes. In contrast, training a network on longer videos can capture more comprehensive, global knowledge across different scales, particularly when the videos share a common theme, such as eating a meal, where people follow behavioral patterns from the beginning to the end.

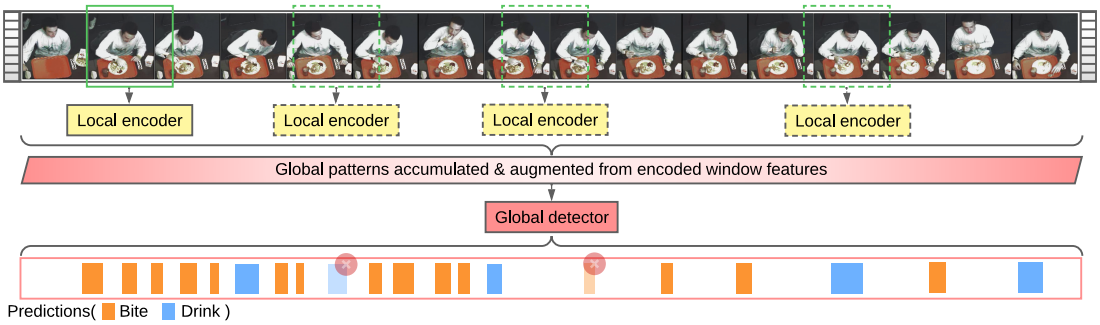
1.4.2 Challenges and Contributions in this Research

We propose a new pipeline, as shown in Figure 1.1, designed to analyze full-length videos from a global perspective. In our benchmarking efforts, we compare our approach against the same methods previously examined by Rouast et al., which demonstrated superior performance, namely SlowFast and CNN-LSTM [75]. Additionally, we include X3D [26], a more recent model known for delivering state-of-the-art (SOTA) performance in generic activity recognition. In the implementation of our pipeline, we adapt CNN-LSTM from Rouast et al. and X3D [26], recognized as SOTA models in the 3D-CNN and RNN categories, respectively, as the window-based backbones.

The computational cost associated with directly building a model to analyze an entire meal video is substantial. We evaluate the computational savings of our approach by comparing a hypothetical full-video analysis in one stage, which utilizes ResNet-34 as the backbone, to our two-stage approach. Both approaches assume training on the Clemson Cafeteria dataset using two Nvidia V100 GPUs, without accounting for the warm-up training phase on other datasets. The forward/backward pass size for processing a single frame with ResNet-34 is approximately 60 MB, which is manageable when analyzing a brief window of typically tens of frames. However, a meal-length video in the Clemson Cafeteria dataset could contain as many as 12,000 frames when sampled at 5 Hz, resulting in a forward/backward pass size for the spatial encoding section alone of about $60 \times 12,000 = 720$ GB. Including temporal components would require even greater computational



(a) Standard video learning via sliding window



(b) Proposed meal-length video learning

Figure 1.1: Standard and proposed video gesture detection methods. The window-based model may struggle with short and fake actions, such as misinterpreting moving hands towards the mouth as a drink or bite gesture, as shown in the figure. In contrast, the global-view model considers the entire meal-length video content and other gestures, allowing it to accurately exclude more fake actions and provide more reliable results.

memory for training. Additionally, the training time for this hypothetical one-stage model will be huge. X3D-L [26] has a forward/backward pass size of 43 GB and takes approximately 118 hours to train on *only* 16 frames at 6 Hz. In contrast, our two-stage approach takes approximately 240 hours to train on *all* 12,000 frames at 5 Hz while outperforming previous works. Thus, our approach enables full-length video analysis with significantly less computational complexity.

The scarcity of available video data also makes it difficult to training a network with full-length videos. Training a video classifier typically necessitates thousands of samples [111], which implies an equivalent number of videos since one video typically represents one full-length sample without additional augmentation. Yet, acquiring such a large volume of ground truth videos is exceedingly time-consuming, with annotating every frame in a 15-minute meal video taking approximately one hour [88]. As a point of reference, the two largest public datasets for intake gesture detection, OREBA dataset and Clemson Cafeteria dataset, comprise only 100 and 486 meal-length videos [77,97]. This limited availability of data poses a significant challenge for training a meal-length video classifier.

To overcome the shortage of video data, we introduce an innovative data augmentation technique during the preparation of samples for training the second network. We observed that the output features of a window-based network (local encoder) exhibit volatility between multiple independent training iterations or when using different frame offsets within a window. This volatility results from random processes such as parameter initialization and data batch variations, as well as from varying forward and backward distances. Leveraging this model volatility, we generate a substantial volume of meal-length feature patterns for training our second network (global detector).

The novelty of this work can be summarized as follows:

1. We propose the first effective two-stage framework for analyzing full-length meal videos to detect intake gestures.

2. We incorporate two state-of-the-art models as window-based backbones and show significant performance improvements following the application of our global analysis framework, as evidenced by improvements in F1 scores and model stability across diverse training runs and individuals.

3. Our experimental findings underscore the effectiveness of full-length pattern analysis on meal videos, surpassing the performance of several window-based benchmark methods.

Detailed methodology, experiments and results are included in 4.

1.5 Novelty

We propose a new approach to conduct a holistic analysis of an entire recording as a single sample. Our approach employs a two-stage methodology, where the first stage employs a sliding window to learn local-level features. In the second stage, a global detector analyzes the first stage results to learn global patterns.

A major advantage of our two-stage approach is the opportunity it provides for *large-scale data augmentation* between the two stages. We propose a novel method that utilizes the uncertainty in the first-stage analysis to generate multiple feature-level representations of the entire recording. The variability in these augmented patterns comes from differences in how the first-stage model interprets the data through repeated iterations of training. This can be contrasted with traditional methods in which random noise, data flipping and rotating are used to create augmented data [13].

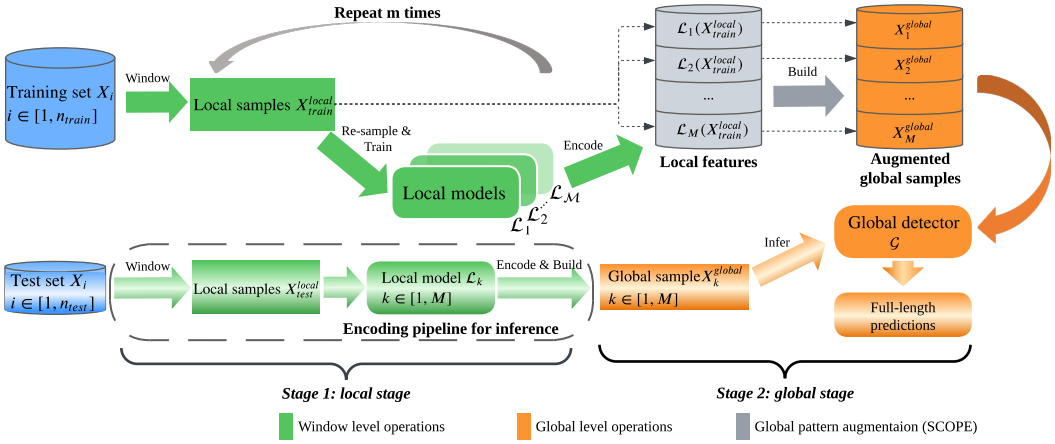
We demonstrate the effectiveness of our pipeline in dietary-monitoring use cases with varying activity lengths, recording durations, and sensor modalities. We also present ablation experiments to show the advantage of our global pattern augmentation over standard augmentation.

The novelties of this dissertation can be summarized as follows:

- We define the problem of sparse (brief or infrequent) activity recognition over a long period of time.
- We develop a two-stage framework for capturing global context in full-length sensor recordings.
- We propose a novel augmentation method to diversify global samples and aid in global detector training.
- We evaluate our framework and augmentation method across completed dietary-monitoring use cases, and provide a generalized meta-framework for additional sparse-event domains.

1.6 General Framework for Sparse Events

This section summarizes sparse-event recognition in lengthy recordings under limited dataset sizes. The target activities may be sparse because they occur infrequently (few instances per record-



(a) Two-stage sparse-event framework with SCOPE augmentation

Figure 1.2: Meta-framework overview used in this dissertation. Stage one encodes window-level evidence, and stage two models full-recording context. SCOPE builds augmented global samples from local-model variability to improve robustness when labeled long-recording datasets are limited.

ing) or briefly (short-duration bursts). In both cases, the total target-event duration is minor relative to the full recording length.

1.6.1 Framework Overview

Figure 1.2 illustrates the general framework. The framework is motivated by application scenarios such as meal detection from day-long wrist IMU recordings, bite/drink detection from meal-length videos, sleep-stage detection from overnight EEG, and speech/music segmentation from long-form audio. It uses a two-stage pipeline to combine local evidence and global context. In stage one, a window-based encoder predicts local probabilities from short segments. In stage two, the full sequence of stage-one outputs is analyzed as one recording-level sample to learn global temporal patterns. To improve robustness with limited data, the framework includes a novel global-pattern augmentation between the two stages, called SCOPE (SynthetiC glObal Pattern augmEntation). By leveraging stage-one model variability and offset-induced uncertainty, we generate multiple recording-level feature traces without requiring additional manual annotation.

1.6.2 Empirical Motivation

Traditional activity-recognition pipelines predominantly use window-based analysis: a sliding window traverses the recording and predicts per-window activity likelihood [14, 18, 35, 94, 108]. This paradigm is effective for local pattern learning, but it limits the temporal field of view because larger windows quickly increase model complexity and computational cost.

Our earlier dietary-monitoring studies showed that sparse events often follow recording-level temporal structure that local windows cannot reliably capture. In both day-length IMU data and meal-length video data, modeling global context reduced false detections and improved reliability.

These observations suggest a broader formulation: many sparse-event tasks share the same structure of long recordings, minority target classes, and constrained labeled datasets. This motivates a unified framework that keeps local-window efficiency while adding full-recording context analysis.

1.6.3 Sparse Events in Time Series

Inspired by our research, we define an activity as sparse if it occurs *briefly* or *infrequently*. We define these activities as **sparse events**. Example sparse-event tasks include meal-episode detection from day-length IMU recordings, intake-gesture detection from meal videos, REM-stage detection in overnight sleep recordings, and speech/music segmentation in long-form audio.

This dissertation addresses the problem of sparse event detection in long recordings.

The classic approach to activity recognition is to apply a local detector in a sliding window across a recording [14, 108]. Existing methods predominantly rely on window-based analysis, where a sliding window traverses the sensor data recording, predicting the likelihood of the activity at each instant [35]. Training involves segmenting the recordings into small pieces containing the target activity and non-activity segments [18, 94].

In our scenario, the sparsity of target events presents a serious class imbalance problem. This challenge is further compounded by the nature of real-world behavior. Activities are rarely performed in isolation; they are interleaved with other tasks, exhibit substantial variability across individuals, and are frequently interrupted by unrelated actions. The scarcity of positive samples makes it difficult for standard models to learn robust and generalizable features. Furthermore, collecting and annotating such sparse events is labor-intensive, which limits the availability of large,

high-quality labeled datasets. A long recording of human activities may contain only a few positive events.

In this dissertation, we propose a framework that incorporates the full-recording context to improve the detection of sparse events.

1.6.4 Global Patterns in Human Activities

During our earlier research on eating gesture detection from video recordings, we observed several common tendencies across individuals. For instance, food preparation activities such as cutting and stirring are typically followed by the intake of food, leading to mastication [73]. Bite gestures, indicative of food intake, tend to occur more frequently than drink gestures. Beverage intake often happens more toward the end of a meal, informally referred to as "washing down food", and intake tends to slow as the person eating becomes satiated. These contextual cues within a meal-length timeframe hold the potential for enhancing the recognition of intake gestures.

We then theorize that many human activities that occur under a specific theme, such as eating-related gestures during a meal, inherently carry sequential context that can be harnessed to improve recognition over extended time periods.

One approach to recognizing rare activities is to consider them anomalies [60]. For instance, a defect during manufacturing can be considered an anomaly [6]. Another example is an occurrence of theft or violence during video surveillance [59,92]. The basic idea in these scenarios is to recognize the events as outliers instead of explicitly learning their patterns during training [60]. This makes sense for rare activities that may occur randomly, or that are difficult to record due to ethical reasons, e.g., violence or medical anomalies [89]. In contrast, we are interested in rare activities that have global context over a long-term span of time, meaning that (a) they occur naturally and commonly enough to facilitate recording, so that (b) they can be explicitly modeled during classifier training. The main difficulty is acquiring a dataset of sufficient size to support the training process.

In this dissertation, we explore modeling global patterns via long sensor recordings in a practicable way. Across our completed studies and framework development, we demonstrate the advantages of leveraging global contextual information.

1.6.5 Challenges in Modeling Global Patterns

One of the biggest challenges in recognizing sparse activities using global context is data scarcity. As opposed to a window-based approach, in which each recording provides a multitude of samples (different windows) for training and testing, our global analysis approach uses an entire recording as a single sample. The labor-intensive process of collecting and labeling long-duration recordings at the datum level limits the size of available datasets [113]. For instance, the Sleep-EDF dataset, a widely-used benchmarking dataset in sleep stage detection, consists of 39 and 153 whole-night healthy recordings across its two published versions [53]. Another example is the Ego4D dataset of 1-10 hr recordings of activities of humans during daily living; described as "massive", and involving 14 different research teams to collect, it still only contains 923 unique participants [33].

Another major challenge in modeling global context, particularly for high-dimensional data such as video, is computational cost. Building an end-to-end model that processes an entire long recording can be prohibitively expensive. To quantify this, we consider a hypothetical 30-minute video analysis task using ResNet-34 as a backbone model. A forward/backward pass for a single frame through ResNet-34 [40] requires approximately 60 MB of memory. While this is feasible for short clips of tens of frames, a meal-length video from the Clemson Cafeteria dataset may contain up to 12,000 frames at a 5 Hz sampling rate. This implies that the spatial encoding alone would consume $60 \times 12{,}000 = 720$ GB of memory. Including temporal modeling components would further increase the memory footprint, making such an approach infeasible for typical GPU resources.

In our proposed framework, we develop a two-stage architecture to reduce computational cost and introduce a novel global pattern augmentation method to address data scarcity.

1.6.6 Data Usage for Time Series Analysis

A fundamental challenge in time series analysis is transforming continuous, unsegmented data into units suitable for classification, detection, or forecasting. In this section, we introduce procedures for transforming raw sensor data recordings into samples for modeling.

1.6.6.1 Intra-Window Modeling

The most widely adopted preprocessing strategy is the sliding window approach, which involves extracting overlapping or non-overlapping fixed-length segments from the continuous signal.

Each window is then treated as an individual sample for downstream modeling. It provides a general way to handle variable-length input, facilitates mini-batch training, and decouples model architecture from the total duration of the input sequence.

Choices of window length and stride determine both the temporal resolution and the computational cost of the system. Shorter windows capture finer temporal details but may miss broader contextual patterns, while longer windows include more temporal context but risk diluting transient signals. Likewise, smaller strides increase overlap between windows, improving temporal granularity at the cost of redundancy and increased processing time.

The window-based framework is widely used across domains, from detecting human activities from wearable sensor data, to classifying videos. It allows for the use of standard supervised learning techniques, where each window is assigned a label, typically inherited from the majority or center of the window.

Representative window settings in prior work include 6-minute windows for meal detection from wrist IMU [86], 2-second windows for video action recognition [26, 27, 75], 30-second windows for sleep-stage classification [24, 82, 95], and 8-second windows for speech/music detection [49].

However, the window-based framework introduces several limitations. One key issue is label ambiguity: a single window may contain a mixture of relevant and irrelevant activity, particularly when transitions occur within the window. This can lead to noisy or inconsistent supervision, especially near event boundaries. Furthermore, the limited temporal field of view inherent to fixed-size windows restricts the model's capacity to recognize higher-order or long-range temporal patterns.

1.6.6.2 Inter-Window Modeling

The limitations of the single window framework motivate the development of architectures that can integrate local information over longer temporal scopes while maintaining computational efficiency.

To address this, some researchers have explored extending the receptive field across multiple windows. Rather than making predictions based on isolated segments, these inter-window approaches aggregate information from a sequence of adjacent windows, enabling the model to incorporate longer temporal context. However, such methods often face challenges arising from data scarcity.

In the domain of sleep stage detection, Seo et al. [82] proposed an approach that leverages

long short-term memory (LSTM) networks to capture inter-epoch information across consecutive 30-second windows. Their method improved recognition accuracy by incorporating temporal context across longer durations. However, the performance gains diminished as the number of input windows increased, a phenomenon they attributed to insufficient training data. In comparison, Eldele et al. [24] demonstrated superior performance using a single 30-second window, which highlights that the SleepEDF-2013 dataset was too small to support effective long-range temporal modeling.

In video-based action recognition, researchers have similarly attempted to incorporate extended temporal context. Yang et al. [112] introduced a collaborative memory pool shared across clip segments extracted from the same video. This design enabled feature sharing across segments and enhanced the model's ability to capture extended temporal structure. Tang et al. [96] employed a global attention mechanism, using long-range attention features to enhance window-based networks. Their experiments processed sequences as long as 768 frames. However, both approaches were limited by computational constraints, which restricted the length of frame sequences used as input. Interestingly, this constraint also mitigated the data scarcity issue, as long recordings could be subdivided into many overlapping or independent segments suitable for training.

Despite these innovations, the limited size of available datasets remains a key challenge in human event detection tasks. For example, the widely used Sleep-EDF dataset includes only 39 and 153 whole-night recordings in its two primary versions [53]. Similarly, the CAD dataset, one of the largest for free-living meal detection, comprises just 354 IMU recordings. The labor-intensive process of collecting and labeling long-duration recordings at the datum level has imposed constraints on the size of available datasets.

In this dissertation, we focus on extending the effective field of view to entire recordings when modeling global context, while addressing computational and data limitations through a more efficient architectural design and a novel data augmentation method.

1.6.7 Machine Learning Models for Time Series

In this section, we introduce popular machine learning models for time-series analysis and their relationships with our research.

1.6.7.1 Statistical Models for Time Series

Before the rise of deep learning, statistical models were the dominant approach for time sequence modeling. Among them, Hidden Markov Models (HMMs) [72] played a central role in applications such as speech recognition, bio-signal analysis, and human activity detection. HMMs assume that the observed signal is generated by a sequence of hidden states, each associated with a probability distribution over the observations. This enables modeling of both stochastic state transitions and probabilistic observations, particularly useful in temporal human motion and physiology.

In our research domain of free-living dietary monitoring, Shen's dissertation [87] from our lab offers a significant example of applying HMMs to real-world sensor data. Shen developed HMMs to recognize complex behaviors such as bites, drinks, utensiling, and rest, from wrist motions during eating, using accelerometer and gyroscope data from a watch-like device. Shen demonstrated that HMMs could benefit from incorporating demographic and meal context (e.g., utensil type, food, participant handedness), and achieved 86.4% and 91.7% on detecting all gestures and intake gestures without gesture history.

Compared to simpler statistical models like ARIMA [8], which assume linear dynamics and stationarity, HMMs offer a flexible framework for modeling temporal events like human gestures or biosignal segments. However, HMMs also face limitations: they assume conditional independence of observations given the current state, and they are limited in capturing long-range dependencies unless explicitly extended (e.g., HMM-N or hierarchical HMMs).

These insights motivate the shift toward deep learning-based models, which offer more flexibility and better scalability when modeling noisy, high-dimensional, and long sequences, a theme we explore in this dissertation.

1.6.7.2 Deep Learning for Time Series

Deep learning has emerged as a powerful paradigm for modeling time sequences, with greater flexibility and learning capacity than classical statistical models. Current deep learning models can be divided into three categories based on their main components, which are introduced as follows.

1.6.7.3 Temporal Convolutional Networks (TCNs)

Convolutional networks (CNNs) have been a basic class for deep learning models. They are primarily for computer vision tasks, but can be easily adapted to temporal convolutional networks (TCNs) for time series data [3]. By applying stacked convolutions with dilated or causal filters, and adding one kernel dimension as the time axis, these models capture patterns over local temporal windows while maintaining the ability to scale to long sequences. TCNs are suitable for temporal patterns that are short, repetitive, or position-invariant.

In our research domain of free-living dietary monitoring, Sharma's dissertation [83] from our lab applied convolutional neural networks (CNNs) for wrist motion analysis in meal episode detection. Sharma proposed a CNN-based architecture that operates on raw inertial signals and uses temporal convolutional blocks to extract salient features related to eating, such as cutting or manipulating food, preparing food for consumption, and resting between ingestion events. Unlike earlier handcrafted features or statistical models, the CNN model learned motion signatures directly from the signal, and achieved 89% true positive rate with 1.7 false positives per true positive on detecting meals from the Clemson All-Day (CAD) dataset.

1.6.7.4 Recurrent Neural Networks (RNNs) and Variants

Recurrent neural networks (RNNs) were among the earliest deep models designed for sequence data. They process input in a recursive manner and maintain hidden states that summarize information up to the current time step. However, standard RNNs suffer from vanishing/exploding gradients, which limits their ability to model long-term dependencies.

LSTM (Long Short-Term Memory) [43] and GRU (Gated Recurrent Unit) [16] models address these limitations with gated mechanisms, which enable more stable training. Both models are capable of learning temporal dependencies over longer sequences and have demonstrated effectiveness in a wide range of applications, including speech recognition, wearable sensing, and biomedical signal analysis.

RNNs treat input as flat vectors at each time step, and therefore cannot learn from structured features within a single timestep, such as the spatial information present in individual frames within a video.

To address these limitations, local feature extractors are always combined with RNNs. For

example, CNN-LSTM [20] models that combine convolutional and recurrent layers have been widely used for video-based recognition tasks. In these models, convolutional layers are used as a front-end to extract spatial features from raw input at each timestep, such as short windows of accelerometer signals or video clips, before feeding the extracted feature vectors into LSTM layers for temporal modeling. This combination leverages the pattern recognition capabilities of convolutional networks with the sequence modeling strength of recurrent units.

In this dissertation, RNNs play a key role. For example, CNN-LSTM [75] architecture is used in the case study of video-based eating gesture detection. CNNs are first applied to spatially encode frame features, which are then aggregated over time using LSTM layers to model temporal dynamics. This layered approach enables the model to combine local precision with broader sequence understanding.

In addition, the global detector introduced in the methodology section also applies LSTM to learn global patterns using noisy data of moderate size.

1.6.7.5 Attention Mechanisms and Transformer Models

Transformer [105] architectures, originally developed for natural language processing, have increasingly been applied to time series modeling. Their key innovation lies in the attention mechanism, which allows each time step to dynamically attend to others and enable the model to capture dependencies across arbitrary temporal distances. Unlike RNNs, transformers are parallelizable in computation, which contributes to their efficiency and scalability in high-resource settings.

In time series applications, transformers have been adapted for classification, forecasting, and event localization tasks [57,110]. These models have shown strong performance on many benchmark datasets, particularly where the data is dense, well-segmented, and large-scale.

Despite their theoretical appeal, transformers have limitations when applied to real-world sensor data. Their reliance on large training datasets, sensitivity to label sparsity, and relative lack of robustness to noise pose challenges for many behavioral sensing tasks. In our sparse event detection work, particularly in free-living environments where labeled events are sparse and the background noise dominates, transformers were not used as the global reasoning component. Instead, we relied on a simpler architecture, LSTM, that generalizes better with limited data and offers greater interpretability.

However, we found that transformers can be effective when used in more controlled settings

as local sequence encoders. In our extension studies on sleep stage classification from EEG recordings, we employed a state-of-the-art (SOTA) attention-based model module to encode fixed-length signal windows before aggregating them for global-level inference. In this context, where events (e.g., sleep stages) are dense and well-annotated and the signal structure is clean, the use of attention mechanisms led to performance improvements over standard convolutional encoders.

Chapter 2

Detecting Eating Episodes From Wrist Motion Using Daily Pattern Analysis

2.1 Methods

Figure 2.1 shows an overview of our method. The input is wrist motion data captured by accelerometer and gyroscope sensors, such as those embedded in a smartwatch. During the first stage, this data is analyzed using a sliding window classifier to calculate $P(E_w)$ which is an estimate of the probability of eating at each point throughout the day, based only upon the local window of time. During the second stage, this entire day-length probability sequence is then analyzed all at once, by what we call the daily pattern classifier. The sampling resolution of the input to the daily pattern classifier is reduced to limit the computational complexity in the neural network. Then with our novel augmentation technique, sufficient day-length sequences are generated to train the second stage neural network. We denote the second stage output as $P(E_d)$ which is an estimate of the probability of eating at each point throughout the day, based upon analysis of the entire day. This output is then post-processed to detect eating episodes (meals/snacks). The following sections describe each of these steps in more detail.

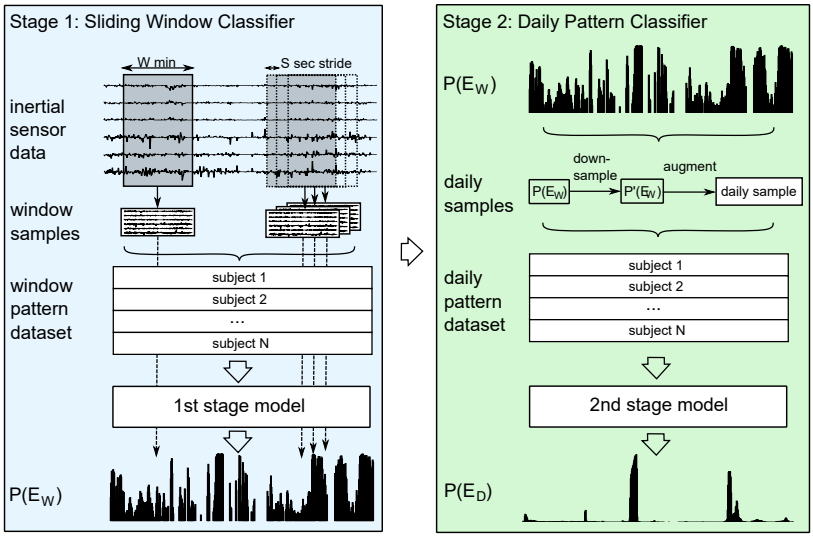


Figure 2.1: Overview of our new classifier that uses day-length context to detect when eating occurred. In stage 1 we analyze a sliding window of raw sensor data to calculate the probability of eating $P(E_w)$ within the local window. In stage 2 an entire sequence of $P(E_w)$ is analyzed as a single sample to calculate the daily probability of eating $P(E_d)$. We augment the original window pattern dataset to a much larger daily pattern dataset using a novel technique that leverages model volatility, to provide sufficient data to train the stage 2 model.

2.1.1 Sliding window classifier

As shown in figure 2.1, a sliding window is used to create samples from wrist motion data. These window samples are used to train a window-based eating detection model. The model needs to be designed to output a probability of eating $P(E_w)$ ranging from 0 to 1 indicating the likelihood that the provided input window sample contained eating. By concatenating the outputs from adjacent windows, a continuous $P(E_w)$ sequence is generated for an entire day, as shown at the bottom of the left side of figure 2.1.

Many previous works could be used for this stage 1 classifier [22, 42, 55, 76, 85]. Note that when building a day-long probability sequence, a probability of eating is required for each window. Thus for those models detecting individual intake events (for example, bites [42,55,76]), a probability of eating could be derived by counting detected intake events within a window. On the other hand, for those models detecting eating episodes (for example, eating vs. non-eating [85]), the probability of eating is based on a majority vote of the window: if more than half of a window contains eating, the window is marked as eating in the ground truth.

During our experiments, we adopt the state-of-the-art eating episode detection method

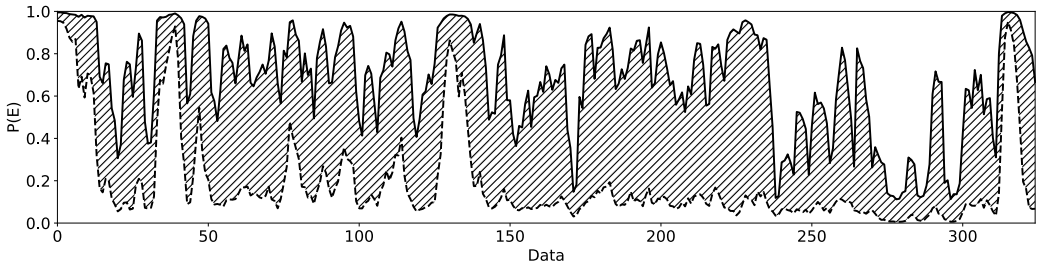


Figure 2.2: Difference (shading) between the minimum (lower dashed line) and maximum (upper solid line) estimates of $P(E_w)$ from 565 retrainings of the sliding window model.

described in [85] to detect the local probability of eating within windows. This method uses a neural network to analyze a window of wrist motion data and is trained to directly output the probability of eating. The network includes three 1D convolution layers, a pooling layer and a dense layer. The window length and stride are 6 min and 15 sec. All model and window parameters were optimal choices according to the original work [85].

2.1.2 Data augmentation

The CAD dataset contains 354 day-length recordings, and the FreeFIC dataset contains 22 day-length recordings. To train our daily pattern classifier we developed a novel data augmentation process that expands CAD by 565x to 200,010 day-length samples, and FreeFIC by 910x to 20,020 samples. Our process takes advantage of the natural variability of output from a neural network each time it is retrained [31]. The variability is caused by differences in the learning process each time the model is retrained, as well as differences in data selection. In general, eating occurs about 5% of the time compared to non-eating (about 1 hour total per day) [22]. During model training, non-eating data is randomly undersampled to balance against eating data. Therefore each time the model is trained a different subset of non-eating data is used. Together with the natural variability in training a neural network, a slightly different output is produced each time the model is retrained. Figure 2.2 shows an example of the effect of this natural variability. The figure plots a single day-length recording after repeated training and testing, showing the maximum and minimum $P(E_w)$ output by the model at each point during the day. It can be seen that repeated training of the model yields significant variation in estimates of $P(E_w)$ at each moment in the day.

Based on this finding, we are able to generate sufficient different day-length samples of $P(E_w)$ from each actual daily recording. During our experiments, we repeat training the window-

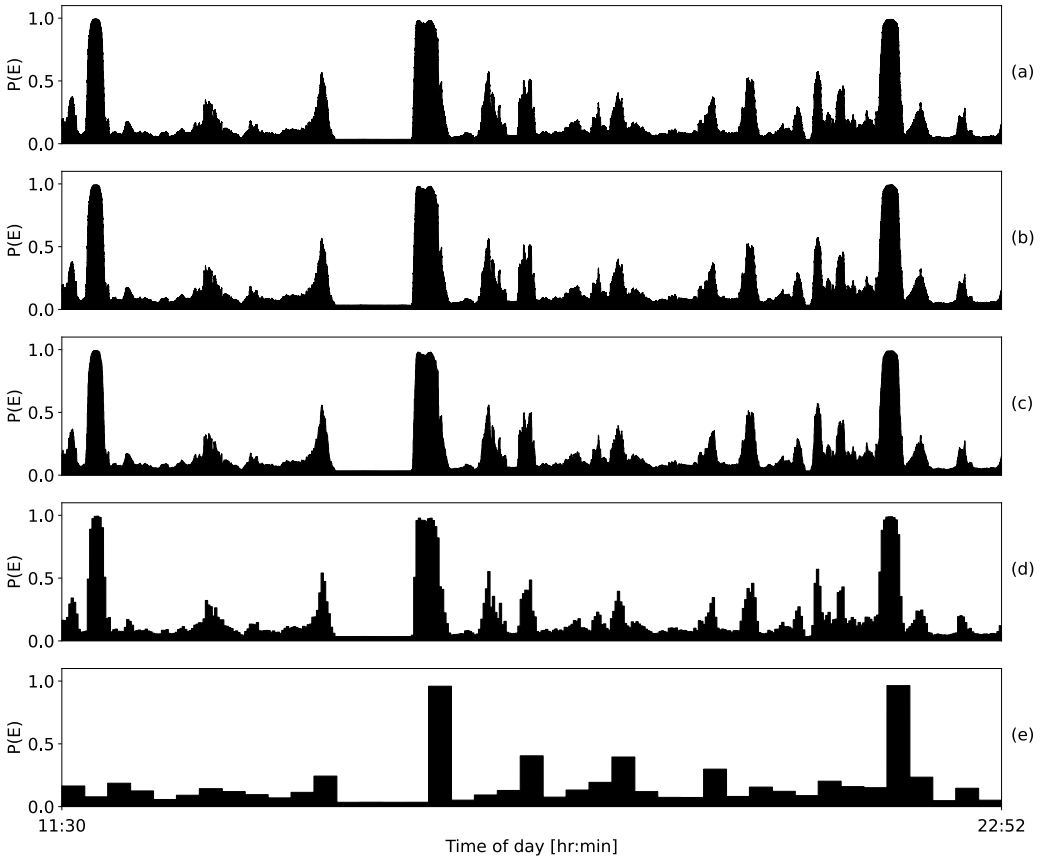


Figure 2.3: Effect of sampling interval for $P(E_w)$ on a daily sample: (a) $\frac{1}{15}$ sec, (b) 1 sec, (c) 10 sec, (d) 100 sec, (e) 1000 sec.

based model 565 times on CAD dataset, and therefore obtained 200,010 daily samples from the daily recording dataset. The same augmentation process is performed on FreeFIC dataset, yielding 20,020 samples.

2.1.3 Daily classifier

A challenge in analyzing a day-length recording is that there is a large amount of data in a day. A classifier that considers all this data simultaneously would thus have a very large number of parameters and suffer from long training and inference times. To reduce the data complexity of a daily sample, we raise the temporal sampling interval by increasing the sliding window stride for computing $P(E_w)$ from 1/15 sec to 100 sec (1 datum every 100 sec). This gives us a daily resolution of appx 1.7 min, which is reasonable when calculating whether someone was eating or not for every moment throughout an entire day.

The effect of the analysis of $P(E_w)$ at this resolution can be visualized. Figure 2.3 shows an example of the same day-length $P(E_w)$ at different resolutions. Each panel from (a) to (e) increases the sampling interval by an order of magnitude. It can be seen that the overall shape of the day-length $P(E_w)$ does not start to noticeably change until (e) which is appx one sample per 15 minutes. Panel (d) shows our chosen resolution. If the first stage classifier is considered as an encoder, then it can be thought of as extracting information from the complex wrist motion data that is useful for detecting *daily* eating. Therefore, the second stage network does not necessarily require a large capacity to analyze daily samples.

During our experiments, we build our second stage network using a single bidirectional RNN layer, and a dense layer applied to each timestep simultaneously. Recursive neural networks (RNN) have strength in processing time series data [15]. The bidirectional layer enabled the network to capture patterns both forward and backward. The final dense layer reduced the RNN layers output to one probability value within range [0, 1] per timestep. The network takes a 1-channel daily sample and output a sequence of probabilities of eating $P(E_d)$. The timesteps of input and output were matched.

A grid search is used to find the optimal number of units U in each layer and the type of layer (LSTM [43] and GRU [15]) that offered the best performance. Powers of 2 from 8 to 256 are tested since these are common in RNN design. The GRU layers outperform the LSTM layers by 1-2% for every number of memory units except 128. GRUs have been shown to pick up on less prevalent patterns which may explain this slight performance difference [34]. Based on these results, we chose $U = 16$ units with a GRU layer.

Z-score standardization is applied to $P(E_w)$ using the mean and standard deviation from training folds.

Since each daily sample is with a different length, all are were zero-padded to the maximum length of 850 (equal to 23.6 hours) before being fed to the network. Every padded timestep acts as a blank placeholder and does not participate in loss calculation and gradient descent. This trick could also be used to fill in periods of time in which a wearable device is not being worn.

2.1.4 Post-processing

The $P(E)$ output needs to be thresholded for final classification. For the window-based classifier used in this work, a dual-threshold hysteresis approach is used in which the $P(E)$ signal

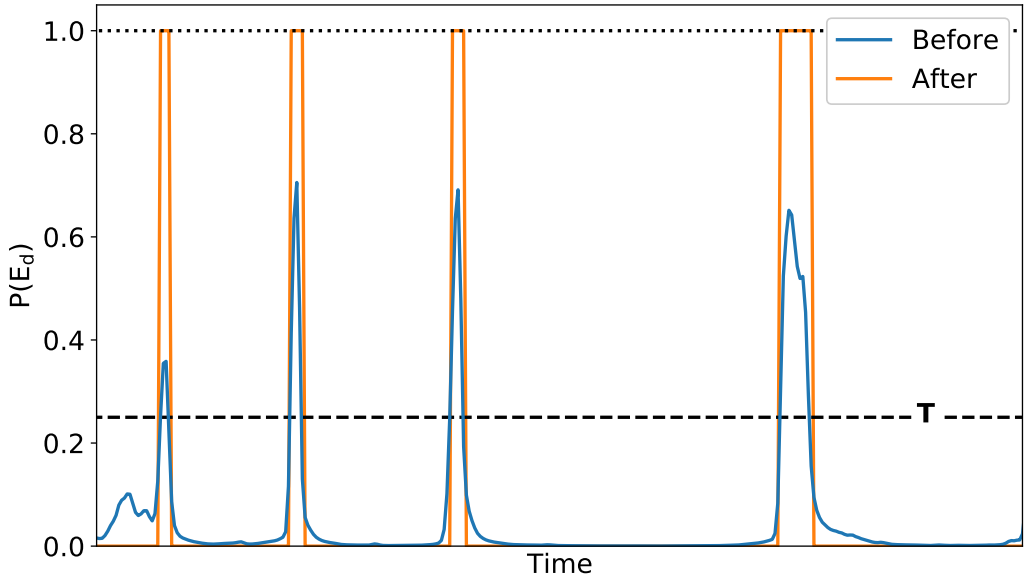


Figure 2.4: $P(E_d)$ signal before and after being processed with the single-value thresholding algorithm ($T = 0.25$).

would have to exceed a starting threshold T_S and fall below an ending threshold T_E to count as a detection [85]. This helps overcome background noise. However, there is much less noise in the $P(E_d)$ signal than in the $P(E_w)$ signal, so we use a single threshold instead. Figure 2.4 demonstrates an example using a threshold of 0.25. We test a range of thresholds to determine its effect on accuracy and report these results in section 2.3.

2.1.5 Model training

Binary cross entropy loss and Adam optimizer are utilized for training models. The learning rate is set to 0.001 and the batch size is set to 128. Each window-based model is trained for 30 epochs and every epoch lasts 500 steps. The daily pattern model is trained for 50 epochs and every epoch lasted 1,000 steps. The train set is shuffled after every epoch. We notice marginal performance improvement in the training accuracy and evaluation results with more training epochs. The model with the best training accuracy is saved within each training phase. All model training and testing are performed on 2 NVIDIA Tesla V100 graphics cards from the Clemson University Palmetto Cluster, a high-performance computing system.

2.2 Experiments and Benchmarks

In this section, we introduce the evaluation scheme, including datasets and evaluation metrics, for our proposed daily pattern classifier. We also detail several benchmark methods we implemented for comparison.

2.2.1 Datasets

We use 3 different datasets in our experiments. The Clemson All-day (CAD) dataset contains wrist motion recordings from 351 participants [86], which makes it the largest dataset of its kind currently available [84]. Each participant wore a Shimmer3 device on the wrist of their dominant hand for an entire day. Two participants recorded multiple days, yielding 354 total recordings. The device recorded 3-axis accelerometer and gyroscope sensor readings at 15 Hz. A total of 4,680 hours of data was recorded. There were no constraints on food types, utensils, or eating behaviors. Participants were asked to press a button on the device at the beginning and end of each meal/snack to provide ground truth times of eating episodes. After data cleaning, a total of 1,133 separate eating episodes were marked, totaling 249 hours of data marked as eating [86].

Bottom-up approaches require data in which all intake gestures (bites) are labeled. This granularity of ground truth requires a semi-controlled environment, for example a cafeteria, in which participants can be video recorded while eating to facilitate labeling intake gestures [77, 88]. This type of ground truth is not available in the CAD dataset as it was collected during everyday life. In this work we use the in-meal OREBA dataset[1] [77] for training bottom-up methods for comparative evaluation. The OREBA dataset includes 100 in-meal wrist motion recordings at 64 Hz and 4790 intake events from 100 individuals.

To evaluate our approach on another independent dataset, we use FreeFIC [56]. It contains 3-axis accelerometer and gyroscope sensor data for 22 in-the-wild sessions provided by 12 participants. The dataset was sampled at a rate of 64 Hz, with a total duration of 113 hours. Participants were instructed to document the start and end moments of their meals to the best of their abilities. Liquid consumption and eating directly with hands were not included in the dataset. Although this dataset is far more limited in size and variability of eating compared to CAD and OREBA, it allows us to perform additional tests of the effect of day-length analysis.

[1] Obtained on request

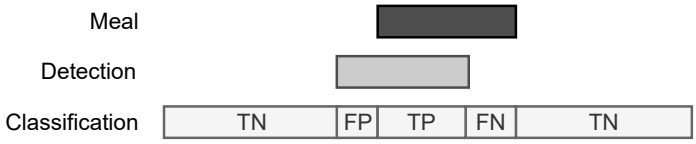


Figure 2.5: Labeling of eating time metrics between ground truth meal and model detection: true positive (TP), true negative (TN), false positive (FP), and false negative (FN)

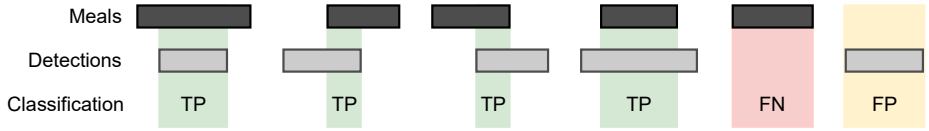


Figure 2.6: Labeling of eating episode metrics between ground truth meals and model detections: true positive (TP), false positive (FP), and false negative (FN)

All experiments were conducted using 5-fold cross validation. The folds were split by subject from each original dataset, rather than by total data. This was done so there was no train/test overlap between samples that originated from the same recording. Within each split, the training window data was balanced by downsampling non-eating windows to the same number of eating windows to prevent classifier bias.

Note that all day-length samples generated from the same original recording were kept together for future training and testing of the daily pattern classifier. In other words, the train and test split for our daily pattern classifier selected from the original recordings and grouped all augmented samples generated from each original day into either the train or test set. This avoids any data leak between training and testing.

During experiments where a bite detection model trained on the OREBA dataset was utilized, the sensor orientations and sampling rates of the target free-living dataset (CAD or FreeFIC, depending on the experiment) were synchronized with the OREBA dataset.

2.2.2 Evaluation metrics

There are two types of evaluation metrics associated with an eating detector: time and episode metrics [84]. Time metrics (figure 2.5) were calculated by comparing the prediction output to the ground truth (GT) for each timestep. A true positive (TP) occurred if the classifier predicted that a timestep was eating and it was also eating in the ground truth. A true negative (TN) was recorded if the classifier predicted non-eating for a timestep that was non-eating in the GT. A false positive (FP) was marked if the classifier predicted eating when the data point was actually non-

stage 1 classifier	stage 1 train	transfer train	stage 2 train	test
Heydarian et al. [42] (2 sec window, transfer learning to 6 min window)	OREBA	CAD	CAD	CAD
Sharma et al. [85] (6 min window)	CAD	-	CAD	CAD
Heydarian et al. [42] (2 sec window, transfer learning to 6 min window)	OREBA	FreeFIC	FreeFIC	FreeFIC
Sharma et al. [85] (6 min window)	FreeFIC	-	FreeFIC	FreeFIC

Table 2.1: Methods and datasets used for testing if our stage 2 day-length classifier yields improved performance.

eating. A false negative (FN) was logged when the classifier missed an eating timestep and instead marked it as non-eating.

Episode metrics (figure 2.6) were calculated by comparing the overlap of model predicted eating episodes and GT eating episodes. In this context, an eating episode is defined as a continuous interval of time classified as eating (i.e. a series of ones). If there was any overlap between the model prediction and the ground truth, the episode was counted as a true positive (TP). A false positive (FP) eating episode occurred when the model predicted an eating episode when there was not one in the GT data. A false negative (FN) was counted when the classifier missed a ground truth event. There is no concept of a true negative (TN) eating episode for episode metrics.

Our evaluation allowed for multiple predicted episodes to match one GT episode. This is primarily due to the way in which participants tend to self-report the ground truth of the start and end of eating episodes during everyday life. Analysis of the CAD dataset found that only 317 (28%) out of 1,133 self-reported meals consisted solely of eating, while the remaining meals were accompanied by secondary activities such as talking, watching television, working, or driving [86]. These secondary activities often cause pauses of minutes to tens of minutes in self-reported meal periods, which can cause two or more valid detections within the GT period [86].

For time metrics, we calculated true positive rate (TPR), true negative rate (TNR), F_1 score, precision, and weighted accuracy (Acc_W). Weighted accuracy is the best overall indicator of time-unit performance because of the class imbalance in the data [84]. For episode metrics, we calculated TPR and FP/TP using episode counts. The number of false positives per true positive (FP/TP) is an indicator of how much the classifier detects false alarms throughout the day.

2.2.3 Evaluation of day-length analysis

In our first set of experiments, we evaluated if day-length analysis could improve the performance of two different stage 1 classifiers on two different datasets. Table 2.1 lists the variations

we tested. These tests were not intended to establish the absolute best performance possible; they sought to determine if adding our stage 2 classifier in each case improved the performance of the original stage 1 classifier on that dataset.

The first stage 1 classifier was a CNN-based model from Sharma et al. that identified if a 6 min window contains eating or not [85]. The second stage 1 classifier was developed using transfer learning to convert the bite detection model from Heydarian et al. [42] to the task of eating detection. Specifically, we first trained the original 2-sec bite detection model on the OREBA dataset using the ground truth for bite moments. We then removed the last dense layer (which in the original model outputs the probability of a bite) and used the remaining part as a kernel to extract bite features from sensor data. The 2-sec kernel was applied to the span of a 6-min window to create a feature map of the same length, with a moving stride of 1 sec. Then, we added 3 CNN layers and 2 dense layers to process the feature map and output the probability of eating. The 3 CNN layers had kernel sizes of 16, 8 and 8, respectively, and filter sizes of 8, 16 and 16, respectively. The stride for each layer was set to 1. The two dense layers had 64 nodes and 1 node, respectively. The layers from the original bite detection model were frozen after finishing training on the OREBA dataset, while the added layers learned to detect eating during training on the CAD dataset. The new model was designed to work on a 6-min window to match the length of the other stage 1 classifier for equivalent comparison.

To process an entire day-length recording, each stage 1 classifier was slid at a stride of 100 sec. The output was thresholded using hysteresis (two thresholds), as in [85]. The high threshold must be surpassed to trigger a meal detection, while all values higher than the low threshold and connected to a value exceeding the high threshold were also included in the detection.

After training the stage 1 classifier, including any necessary transfer learning, all four variations listed in table 2.1 were used to train and test a stage 2 classifier. All these tests used the methods described earlier for data augmentation, training and testing. The stage 2 results were then compared to the stage 1 results to determine if adding day-length analysis had any effect on the accuracy of detecting eating episodes.

2.2.4 Benchmark methods

For the purpose of benchmarking the performance of our new classifier on the problem of detecting eating episodes, we evaluated all methods using the CAD dataset. It contains free-living

Method	stage 1 (window level)			stage 2 (day-level)	
	method	train on	transfer train	method	train / test on
Heydarian et al. [42], Kyritsis et al. [56] (2 sec window)	CNN-LSTM	OREBA	-	Kyritsis et al. meal detector [56]	CAD
Heydarian et al. [42] (transfer learning) (6 min window)	CNN-LSTM	OREBA	CAD	hysteresis meal detector [85]	CAD
Our framework + Heydarian et al. [42] (transfer learning) (24 hr data)	CNN-LSTM	OREBA	CAD	our daily classifier	CAD
Dong et al. [22, 86] (1-min∼30-min window)	Bayesian classifier	CAD	-	-	CAD
Sharma et al. [85] (6 min window)	CNN	CAD	-	hysteresis meal detector [85]	CAD
Our framework + Sharma et al. (24 hr data)	CNN	CAD	-	our daily classifier	CAD

Table 2.2: Methods and datasets used for benchmarking.

recordings from 351 subjects which is far larger than other free-living datasets including the freeFIC dataset with recordings from 12 subjects. We believe that the large size and diversity of this dataset provides the most realistic assessment of how methods could be expected to perform on new data.

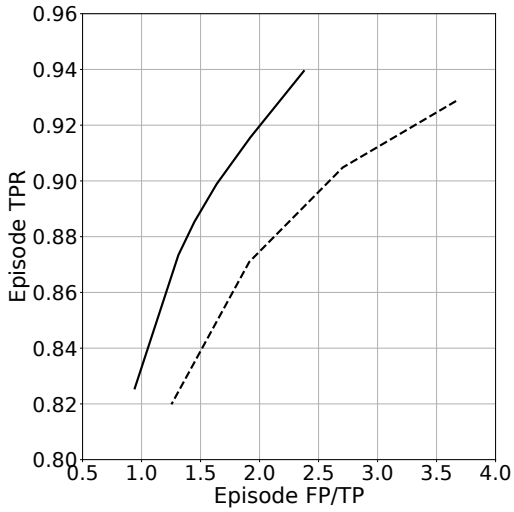
We compared our results against 3 state-of-the-art methods to benchmark our performance. The first method (Kyritsis et al.) is a bottom-up approach that uses a window-based neural network to detect individual intake gestures (bites) and a meal detector to post-process the bite detection results [56]. We adapted the best neural network structure from Heydarian et al. [42] who obtained the best results for detecting bites on the OREBA dataset to date. Specifically, the neural network consists of 4 convolutional layers, 2 LSTM layers, and a fully connected layer. It operates on a 2-sec window and outputs the probability of a bite. The model is applied with a sliding window on a recording, and all output probabilities are fed to a local-maxima search algorithm to identify bite detections. The original authors reported an F1 score of .654 for detecting bites using their model on the OREBA dataset. We tested our reimplementation and found an F1 score of .655 on the same dataset with a near-equal balance of false negatives and false positives, giving us confidence that our implementation was working as intended. We then combined the trained bite detection model with the meal detector as described in [56], and tested the whole pipeline on the CAD dataset. The meal detector works on a binary sequence of bite detections (1=intake, 0=no intake) to detect meal episodes. Specifically, the binary sequence is smoothed using a Gaussian filter to close small gaps between bite groups, then thresholded to filter out sparse detections. The start and end of meal episodes are found by paired consecutive edges detected using a 1D edge filter. Gaps less than 180 seconds between two meal detections and single meal detections less than 180 seconds are removed. The threshold for local maxima when searching for bite detections, and the threshold applied on the

Gaussian filter output for detecting episodes, were chosen to obtain the best possible results.

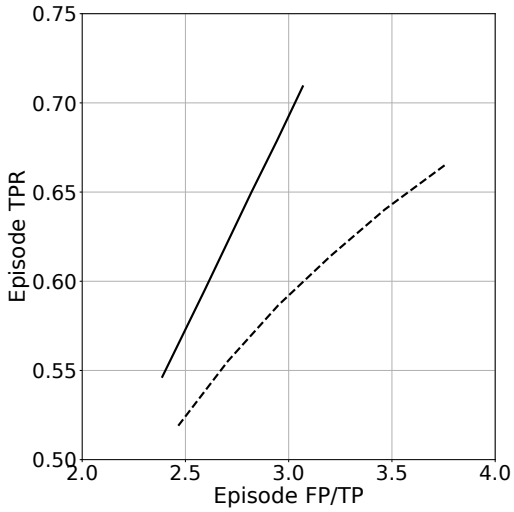
Training the bite detection neural network requires data in which every bite is labeled. This granularity of ground truth is not available in the CAD dataset because it was collected during everyday life. This is a limitation in our ability to test bottom-up approaches. Ideally, a dataset would exist that was (a) collected during everyday life (so as to represent general eating behaviors), (b) has every bite labeled in ground truth (so as to enable training of bottom-up approaches), and (c) contains recordings from hundreds of people (so as to have sufficient diversity to reliably predict future performance). In the absence of such a dataset, the best we can do is train on a large semi-controlled bite dataset (OREBA), and evaluate on the largest available free-living dataset (CAD). However, we recognize that there may be limitations to this approach, as datasets were collected under different conditions. We therefore tested three variants. The first variant was a direct reimplementation as just described. The second variant embedded the bite detector in a 6-minute window and used transfer learning on the CAD dataset, as described in the previous section. This variant allowed us to test the effect of differences between training purely on the OREBA dataset, vs training on both OREBA and CAD. The third variant added our day-length analysis to the second variant, also as described in the previous section. This final variant allowed us to test the effect of adding day-length analysis to the bottom-up approach.

The second benchmark method (Dong et al.) is a Bayesian classifier [22] that is augmented to recognize secondary activities that can obscure motion patterns during eating [86]. We report the previously published results on the CAD dataset [86].

The third benchmark method (Sharma et al.) is a sliding window approach that uses a neural network to analyze a 6 minute window of data to determine the probability of eating [85]. We reimplemented this method with the training settings described insection 2.1.5. Five-fold cross validation was repeated 565 times with the same fold splitting, and the testing results for each fold were obtained by averaging each run. We tested two variants of this classifier. The first variant was a direct reimplementation as just described. The second variant added our day-length analysis, as described in the previous section.



(a) Stage 1 classifier: Sharma et al.



(b) Stage 1 classifier: Heydarian et al.

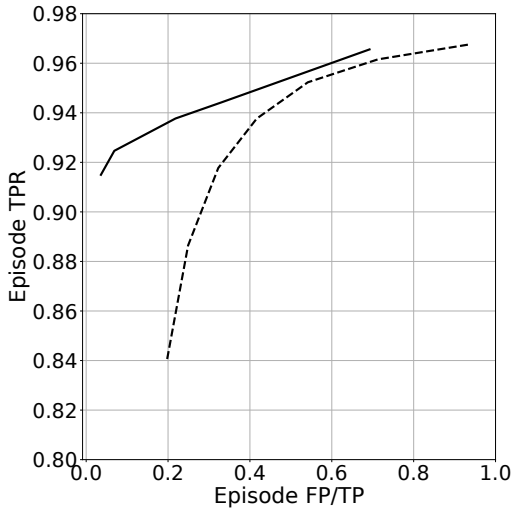
Figure 2.7: Episode detection accuracy of two stage 1 classifiers with (solid line) and without (dashed line) day-length analysis, on the CAD dataset. Both show improved accuracy.

2.3 Results

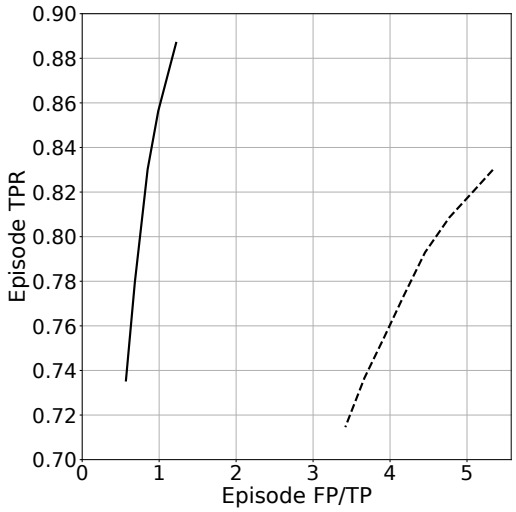
Section 2.3.1 evaluates the benefit of day-length analysis on the detection of eating compared to standard sliding window analysis. Section 2.3.2 compares the results of our new method against state-of-the-art previous works.

2.3.1 Daily vs local window analysis

Figure 2.7 shows the accuracy of two different stage 1 classifiers on the CAD dataset, both with (solid line) and without (dashed line) stage 2 day-length analysis. Accuracies are plotted at multiple post-processing thresholds. In general, a higher threshold triggers fewer detections, resulting in fewer true positives but also fewer false positives. Across a range of thresholds it can be seen that including the stage 2 daily pattern classifier provided higher accuracy compared to the original stage 1 classifier. It consistently detected more true positives with fewer false positives. Figure 2.8 shows the same plots for both stage 1 classifiers on the FreeFIC dataset. It can again be seen that including the stage 2 daily pattern classifier provided higher accuracy compared to the original stage 1 classifier. Collectively, these results show 4 independent tests in which day-length analysis improved the detection accuracy of eating episodes compared to bottom-up approaches that



(a) Stage 1 classifier: Sharma et al.



(b) Stage 1 classifier: Heydarian et al.

Figure 2.8: Episode detection accuracy of two stage 1 classifiers with (solid line) and without (dashed line) day-length analysis, on the FreeFIC dataset. Both show improved accuracy.

only examined a few minutes of data at a time in a sliding window. This provides evidence that diurnal eating habits can be learned by a neural network and that eating detection can benefit from day-length analysis.

There are some relative differences in accuracies between the results in figures 2.7-2.8 that warrant discussion. First, both classifiers had considerably higher accuracies on the FreeFIC dataset compared to the CAD dataset (figure 2.8(a) vs figure 2.7(a), and figure 2.8(b) vs figure 2.7(b)). This is likely because the FreeFIC dataset contains far less variability in eating behaviors than the CAD dataset. FreeFIC participants only ate with a spoon or fork (no hand-based eating, no other utensils) and beverage consumption was ignored. We believe that all results reported on the FreeFIC dataset are inflated compared to the performance that could be expected on a more general dataset, such as CAD. The authors of the FreeFIC dataset had a similar finding when applying their classifier to another dataset called ACE [56]. ACE consists of 25 in-the-wild recordings from 11 subjects, so is also of limited size, but does include eating with hands, beverages, and snacks [63]. The FreeFIC authors' bottom-up approach to detecting eating episodes achieved appx 92% recall and 88% precision on FreeFIC, but dropped to 71% recall and 40% precision on ACE [56]. We believe our new experiment adds to the evidence that small datasets with limited eating behaviors should not be used to evaluate the accuracy of eating monitoring methods [84].

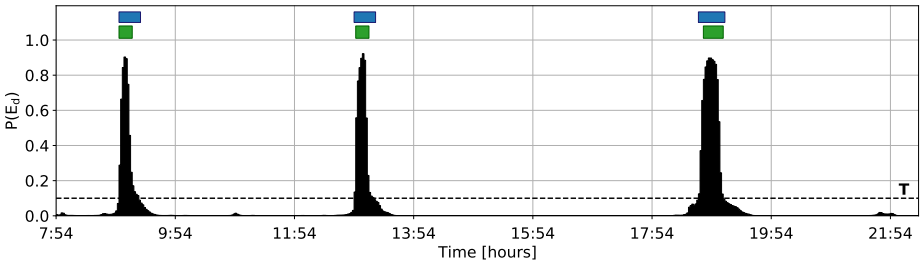
Second, the Sharma et al. classifier had considerably higher accuracy than the Heydarian et al. classifier on both datasets (figure 2.7(a) vs figure 2.7(b), and figure 2.8(a) vs figure 2.8(b)). We believe this is because the Sharma et al. classifier was recognizing eating vs non-eating in a 6 min window, while the Heydarian et al. classifier tried to recognize individual bites within a 6 min window and associate them with eating. In other words, the Sharma et al. classifier took somewhat more of a top-down approach than the Heydarian et al. classifier. As discussed in [85], an eating detection model can learn about non-intake gestures that occur during eating, such as "utensiling" (preparation of food before a bite) and "rest" (lack of wrist motion during mastication). Modeling and recognition of these intake-related gestures is absent in bottom-up bite detection approaches. The classifier presented in this paper takes that idea further to an entire day, where diurnal context produces an additional increase in accuracy in the detection of eating episodes.

Figure 2.9 shows an example recording comparing the result for the Sharma et al. classifier with and without stage 2 day-length analysis. Both classifiers correctly detected all 3 eating episodes with 0 false positives. However, it can be seen that the probability of eating $P(E_d)$ is much cleaner compared to $P(E_w)$. There is a consistently low amount of background noise (probability around 0.2) in $P(E_w)$. When only analyzing a small window of data, many activities throughout the day may have some characteristics that resemble aspects of eating, such as picking up and manipulating things. Confidence in the probability of eating is greatly increased by analyzing a full day of data all at once, as evidenced by the absence of background noise in $P(E_d)$. Figure 2.10 shows another example of daily recordings for these two classifiers. The considerable reduction in background noise is similar to the previous example. It can also be seen that false positive episodes were reduced by our new stage 2 classifier, yielding better overall accuracy.

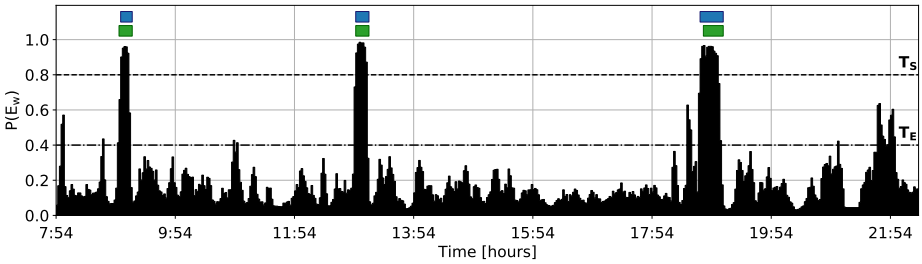
2.3.2 Comparison to previous works

Table 2.3 shows the performance of our method compared to related works on all evaluation metrics. It is important to note that most of the time metrics are bad indicators of performance due to the large class imbalance (5% eating vs 95% non-eating throughout the day) [22]. This is a well-known problem in analyzing the performance of classifiers on detecting eating during everyday life [84]. We therefore focus our attention on weighted accuracy Acc_W and the episode detection metrics.

The bottom-up bite detector method (2-sec window) with grouping of bites to detect

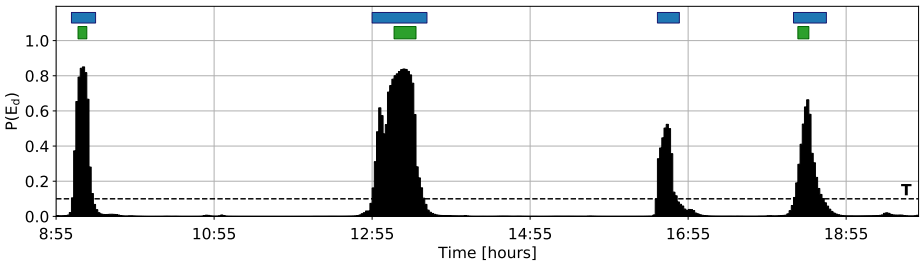


(a) Daily pattern classifier $P(E_d)$ with $T = 0.1$. 3 TP, 0 FP, 0 FN

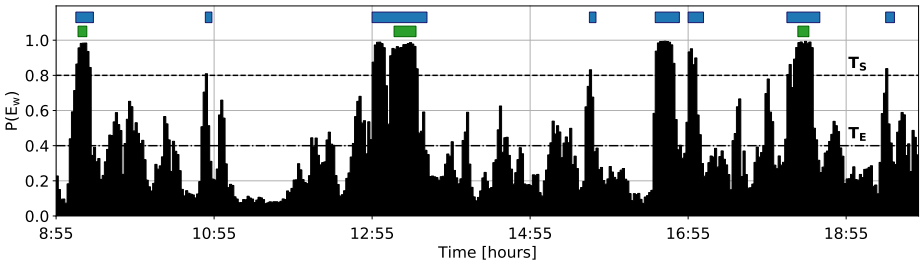


(b) Windowed eating classifier $P(E_w)$ with $T_S = 0.8$ and $T_E = 0.4$. 3 TP, 0 FP, 0 FN

Figure 2.9: Comparison between $P(E_d)$ (our new method) and $P(E_w)$ [85]. Detections shown with blue bars (top) and GT shown with green bars (bottom). Sliding window analysis has a consistent background noise ($P(E_w)$ appx 0.2), while daily pattern analysis is much cleaner. Raw data is from the CAD dataset.



(a) Daily pattern classifier $P(E_d)$ with $T = 0.1$. 3 TP, 1 FP, 0 FN



(b) Windowed eating classifier $P(E_w)$ with $T_S = 0.8$ and $T_E = 0.4$. 3 TP, 5 FP, 0 FN

Figure 2.10: Comparison between $P(E_d)$ (our new method) and $P(E_w)$ [85]. Detections shown with blue bars (top) and GT shown with green bars (bottom). Our daily pattern classifier has much fewer false positive episodes. Raw data is from the CAD dataset.

Method	time metrics					episode metrics	
	TPR (%)	TNR (%)	F_1 (%)	Precision (%)	Acc_W (%)	episode TPR (%)	episode FP/TP
Heydarian et al. [42], Kyritsis et al. [56] (2 sec window)	36	78	14	9	56	59	19.5
Heydarian et al. [42] (transfer learning) (6 min window)	54	65	14	8	60	59	3.0
Our framework + Heydarian et al. [42] (transfer learning)	60	82	25	15	71	71	3.0
Dong et al. [22, 86] (1-min∼30-min window)	76	73	23	14	77	86	3.8
Sharma et al. [85] (6 min window)	76	86	35	23	81	87	1.9
Our framework + Sharma et al. [85]	81	86	38	25	**84**	**89**	**1.4**

Table 2.3: Time and episode metrics comparing related works to our new framework with daily pattern classifier. The CAD dataset is selected for benchmarking because its large size and data diversity makes it the best available to predict future performance on new data. The combination of our framework and Sharma et al. window-based model shows an increase in Acc_W, an increase in episode TP, and a decrease in episode FP/TP, all of which are the best measures reported on this dataset.

eating episodes scored lowest at 56% weighted accuracy. It found only 59% of eating episodes (meals/snacks), with 19.5 false positives for every true positive it found. With transfer learning to CAD, the weighted accuracy improved to 60%, and there was a dramatic reduction in false positive episode detections. This is likely because the types of intake gestures in CAD have more variability than in OREBA, due to CAD being recorded during everyday life while OREBA was recorded in a cafeteria; transfer learning enabled the classifier to better learn this variability. However, the episode TPR remained at 59% even after transfer learning. The third variant, which added our day-length classifier, scored 71% weighted accuracy, and improved the episode TPR to 71%. This shows that day-length analysis can significantly improve the detection of eating episodes for a bottom-up classifier.

The Bayes classifier analyzing small windows (typically 1 min to 10 min, 30 min max) scored 77% weighted accuracy. It found 86% of eating episodes, but yielded 3.8 false positives per true positive. This value is too large for many practical applications. For a person eating 3-4 meals per day, this classifier would trigger 11-15 false positives during the day.

The method using a neural network to analyze a 6-minute sliding window scored 81% weighted accuracy. This is slightly better than what was reported in the original paper describing this method (80% weighted accuracy) [85]. We believe this is because our reimplementation averaged its performance across 565 runs on the CAD dataset, whereas the original paper reported only 1 run.

The variant which added our day-length classifier improved weight accuracy to 83%, which

is the highest achieved on the CAD dataset. It found 89% of eating episodes with 1.4 false positives per true positive. These episode detection metrics are also the highest achieved on the CAD dataset. This shows that daily patterns related to eating can be learned by a neural network and that this knowledge improves the recognition of eating episodes in wrist motion. It is also interesting to note that across all six methods there is a trend of improvement in the performance metrics as the window size increases. This supports the hypothesis that top-down analysis includes more context that can be beneficial for detecting eating, as opposed to the bottom-up approach of looking for individual intake gestures.

2.4 Discussion

In this paper, we introduced the novel concept of analyzing a day-length recording of wrist motion as a single sample to detect eating. We designed a new pipeline that combined an existing window-based classifier with an RNN-based daily pattern classifier for this task. Compared to other state-of-the-art related works, our new method achieved a higher time weighted accuracy, higher true positive episode detections, and lower false positive episode detections. Specifically comparing to a 6 minute sliding window-based classifier, our new daily pattern classifier yielded far less background noise in its determination of the probability of eating $P(E)$; in other words, the classifier had more confidence in its determination of when eating is occurring or not. We believe this is because in a window of a few minutes length, a variety of gestures and wrist motions throughout the day can resemble brief periods of eating. These gestures may include grooming, shaving, brushing teeth, adjusting glasses, touching the face, and even food preparation, among a multitude of others. However, when this data is analyzed within a day-length window, the additional context improves classification accuracy.

The ideas described in this paper could be applied to any wearable designed to detect eating. We demonstrated the value of day-length analysis on wrist-worn devices that measure hand-to-mouth intake gestures [41]. However, the same idea could be applied to earpiece devices that measure motions and sound associated with mastication (chews) [81]; eyeglass and earpiece devices that measure motions and sounds associated with mastication (chews) [81]; and throat-located devices that measure forces and sounds associated with ingestion (swallows) [107]. Any device that relies upon detecting brief physiological events associated with eating will suffer from detecting transient

false positives throughout the day. Our day-length classifier could significantly improve all those approaches.

In our comparative evaluation, we showed the poor performance of a bottom-up approach to detect eating on a large dataset. To our knowledge, this is the first time that this type of evaluation has been performed. Previous works have evaluated multiple bottom-up approaches but only on small datasets consisting of 12 or less subjects [56, 62, 100]. Evaluations on datasets this small tend to overestimate accuracies that could be expected in general everyday life [84]. While it was not a main purpose of this paper to evaluate bottom-up approaches to detecting eating, we hope that our comparative evaluation encourages future works to avoid evaluations on small datasets.

Our new daily pattern classifier requires a full day-length recording of wrist motion for data for analysis, which means that it is not suitable for real-time analysis. However, the classifier can be applied at the end of each day, or across a set of complete days, in order to provide summative information regarding the timing and duration of each eating episode throughout each day.

It is important to note that the sensor data within detected eating episodes could subsequently be reanalyzed to detect individual intake gestures (bites and drinks), and to characterize eating rate. Thus, the top-down approach does not preclude a fine-grained analysis of eating behaviors; in fact it simplifies the computational burden by focusing that analysis only on periods of eating. There are numerous other important research and clinical applications for day-length analysis. First, a day-level classifier could be used to obtain accurate and unobtrusive measures on the timing and duration of eating, thus allowing us to answer important research questions about the health benefits and mechanisms of time-restricted eating interventions [28, 32, 39]. Second, a day-level classifier could be used to enhance gold-standard dietary assessment approaches [102]. For instance, 24-hour dietary recalls rely on an individual to recount all food/drink consumed the day prior [101]. A day-level classifier could improve the accuracy of this approach by providing an end-of-day summary to the researcher or participant to help them remember all their meals/snacks during the recall. Third, a day-level classifier could transform how we evaluate the effects of health-related interventions on diet and eating patterns by enabling reduced-burden data collection over longer periods of time via widely-available wearable sensors in which outcomes are typically evaluated in aggregate over days or weeks at the end of a designated assessment period [64].

Clinically, a day-level classifier could be immensely helpful to reduce the burden of self-monitoring (i.e., tracking all foods and drinks consumed), which is the cornerstone of most behavioral

approaches for changing eating habits to promote health [10]. At a minimum, the daily pattern classifier could be applied in real-time at the end of each day to remind individuals to complete dietary records and assist in improving accuracy via identification of missed eating events or discrepancies in portion size [1]. While a limitation of the day-level classifier is that real-time feedback cannot be delivered directly during an eating episode, the information gleaned from the classifier at the end of the day could still be used for intervention, e.g., providing summative feedback to an individual and facilitating goal-setting for the next day. In sum, the daily pattern classifier described in this study supports measurement of eating patterns in the context of daily life in a way that holds immense value for understanding and improving eating behavior.

We are making our software and models publicly available.[2] The software includes all our source code (for developers) and a GUI-based executable (for end users) that can load and analyze files of wrist motion from multiple different types of devices, and outputs the detected episodes. It also implements a bite detector and reanalyzes detected eating episodes to characterize bite count and eating rate. The software is designed for use by non-programmers, e.g. clinicians studying treatments for obesity.

In future work, we aim to explore a single-stage day-level analysis instead of the two-stage approach described in this paper. The current daily pattern classifier takes input from the output of another classifier rather than directly from the raw wrist motion IMU data. We plan to investigate the possibility of integrating both models, namely the window-based and daily-pattern models, into a single end-to-end encoder-decoder classifier. Furthermore, our top-down approach allows for splitting the eating monitoring problem into two steps: first, detecting periods of eating; second, analyzing those periods to identify characteristics of intake. After identifying periods of eating, a secondary analysis could be performed only on those periods to identify and count individual bite and drink gestures [21, 41, 55, 63]. Future research could explore new ways of combining these types of bottom-up analysis with our top-down approach.

[2] `https://cecas.clemson.edu/ahoover/bite-counter/`

Chapter 3

A New Video Dataset for Recognizing Intake Gestures in a Cafeteria Setting

3.1 Related Work

A few previous works have demonstrated the capability to recognize intake gestures from video camera data [47, 70, 75]. Table 3.1 shows the datasets that have been reported in published studies about intake gesture detection from video, along with our new dataset as a comparison. Some other related datasets of intake gestures, such as FIC [55] and ACE [61], mentioned that video data was recorded, but the video was only used to mark the ground truth of intake gestures in synchronized wrist motion. The video data itself was not used to recognize intake gestures.

As we can see, all current datasets were collected in laboratories with restrictive conditions. The datasets either have a small, controlled set of foods or a relatively small number of subjects, both of which limit the variability in appearances of collected gestures. Cameras were conspicuously close to participants, which could potentially affect the eating behaviors of participants. These restrictions limit the data variability, and make the trained model difficult to generalize to other data. This problem becomes more significant when it comes to developing video-based detectors which rely on neural networks for video analysis and require large amount of data to avoid overfitting. Because of

Table 3.1: Video Dataset of Intake Gestures

Name	Participants [1] (Videos)	Gestures (Types)	Environment	Camera position	Food choices	Availability
OREBA [75, 77]	102 (102)	4279 (Bite, drink)	Lab	On the dinning table and in front of the participant; front view.	lasagna, bread, yogurt, and water	Public[2]
- [47]	28 (84)	2101 (Bite)	Lab	Side of the participant with a 3-feet distance; side view.	Participant choice (total types not reported)	Private
- [70]	12 (52)	2039 (Bite)	Lab	On the shoulder of the participant; over-shoulder view.	66 food and beverage choices	Private
Clemson Cafeteria (ours)	264 (486)	23142 (Bite, drink)	Public cafeteria	On the ceiling of 5 meters height; over-head view.	374 food and beverage choices	Public[3]

[1] In our dataset, one participant contributed one meal of 1-4 courses, and each course was recorded as one video. In other datasets, one participant could contribute multiple single-course meals, and one meal was one video.
[1] See `http://www.newcastle.edu.au/oreba`.
[2] See `http://cecas.clemson.edu/~ahoover/cafeteria/`.

these limitations, the current datasets can only demonstrate proof-of-concept.

3.2 Novelty

In this study, we aim to address the dataset limitations in related research by releasing a new video dataset collected in a more realistic environment. The video data is a part of the Clemson Cafeteria Dataset [45] of which the IMU and scale data were released already[1] [88]. The video data was recorded in a large university cafeteria. Data was collected by 276 participants eating a total of 374 different foods and beverages of their choice [88]. Participants took their meals together with other guests in the restaurant, and had access to all foods provided by the restaurant. The ceiling-mounted cameras were positioned 2-3 m above a single four-seat table and were inconspicuous to diners [45]. The table was located in a normal place in the restaurant with the crowd walking around. A total of 7 camera positions were used, each had different locations at the ceiling or different angles of view. The data collection lasted about three months and covered breakfast, lunch and dinner across the whole day time. This allowed us to inconspicuously observe a large variability of intake behaviors in a natural setting.

To explore the feasibility of detecting intake gestures on our dataset and provide a baseline for future research, we adapted three neural networks which achieved cutting-edge performance

[1]See `http://cecas.clemson.edu/~ahoover/cafeteria/`

on either laboratory intake gesture video dataset or generic large-scale video classification dataset. With our dataset, the best model achieved F1 scores of 0.899 and 0.778 for detecting bite and drink gestures, respectively.

To summarize, the novelty of our study includes:

- We present a new video dataset of intake gestures with more variability than previous experiments, including more participants, more meal choices, and an overhead camera viewpoint.
- We investigated three state-of-art neural network models to show the feasibility and baseline of detecting intake gestures using our dataset.

3.3 Dataset

In this section, we describe our dataset and an enhanced ground truth we created to serve the purpose of video analysis.

3.3.1 Overview

The data used for this research is the video part of the Clemson Cafeteria Dataset [88]. The Clemson University Institutional Review Board approved data collection. The dataset was collected from 276 participants, where each participant ate a single meal consisting of 1-4 courses at the Harcombe Dining Hall at Clemson University. Each course was recorded into a separate video. Up to four participants were recorded simultaneously, each by a different video camera. Four 480p cameras were placed on the restaurant ceiling to record participant activities. During the data collection, some cameras were moved, which led to seven different camera viewpoints. Figure 3.1 shows some recorded frames.

There are 486 total usable videos in the dataset. The participant group consists of 137 females and 127 females, of which 183 were between 18-30 years old, 60 were between 31-50 years old, and 21 were between 51-75 years old when the data was collected. A total of 374 different food and beverage types appeared in their meals, including stir-fried vegetables, shoestring French fries, pasta, water and soda. Utensils included forks, spoons, chopsticks and hands. Containers included plates, bowls, glasses and cups.

Of the original 276 participants recorded, 5 were excluded due to failures on recording

Figure 3.1: Examples of recorded data frames.

Figure 3.2: Example of enhancing gesture ground truth (GT) for a piece of videos. A drink gesture taken with the non-dominant hand has been added. (The participant's dominant hand is left hand. Bite and drink are labeled with yellow and blue colors respectively)

videos or wrist motion data [88]. An additional 7 were excluded in this study because they ate only with their non-dominant hand and thus were completely missing gesture-level ground truth (the original study only annotated gesture-level ground truth for the dominant hand; this is discussed more below). Finally, 2 videos were excluded because the camera was moved mid-recording. These left data for 264 participants and 486 courses (i.e., 486 videos).

3.3.2 Video Ground Truth

The main intent of the original dataset was to measure wrist motion to detect eating gestures [88]. Video was used only to label the ground truth moments of intake in the wrist motion data. Later, a second ground truth was created that identified the beginning and end of each intake gesture, but to save labeling effort, only gestures using the dominant hand (or both hands) were labeled [87]. In this paper we combined these two ground truths to create a new, third ground truth,

that annotates all gestures (regardless of which hand) so it can be used for video-based recognition. Here we first explain the content and limitations of the original ground truth. We then explain our augmentation method and show an example of our enhanced ground truth used for video data.

The first type of ground truth available in the Clemson Cafeteria Dataset is at the moments of intake [45]. It provides a single time index for each moment that food or beverage first entered the mouth (i.e. the moment of intake). This ground truth was recorded for every intake gesture, regardless of which hand (dominant or non-dominant) was used. The type of intake (food or beverage) is also recorded. The original use of this ground truth was to detect intake from wrist motion [88]. This level of ground truth is inadequate for training video-based recognition because it only labels single moments of intake rather than sequences of arm motion related to intake, from which video-based recognition can be expected to make classifications.

The second type of ground truth available in the Clemson Cafeteria Dataset describes gestures. This ground truth provides a span of time for several different types of gesture, including food intake (bite), liquid intake (beverage), manipulating or preparing food for intake (utensiling), and undergoing no motion (resting). This ground truth was also originally used for wrist motion tracking analysis [87]. It provides more detail than the intake-level ground truth, but unfortunately it was only annotated for gestures of the dominant hand where the wrist band was attached to record motion data, in order to save labeling effort. Therefore, while it provides labels for sequences of arm motion, it misses any intake gestures done using the non-dominant hand.

In order to label every frame of video, and to include gestures done using either arm/hand, we created an enhanced ground truth by combining information available from both of the original ground truths. The process begins by copying all bite and drink gestures from the gesture-level ground truth. This labels all frames of all intake gestures taken by the dominant hand. We then augment this by searching the moment-level ground truth for intake done by the non-dominant hand. Any moment found in this ground truth that is missing from the gesture-level ground truth is used to label a new sequence of time. The center point of this sequence is taken as the intake-moment index. The starting point and ending point of this sequence are calculated using the median duration of bite or drink gestures for the entire dataset. An example of this type of enhancement is shown in Figure 3.2. In this example, a drink gesture taken with the non-dominant hand is seen in the video, and appropriately labeled in the enhanced ground truth.

During enhancement, 1,233 bite and 1,335 drink gestures were added to the original gesture

Figure 3.3: Examples of cropped frames captured from different meals.

ground truth. These numbers corresponding to 102 minutes of bite and 35 minutes of drink. In total, the enhanced ground truth of the Clemson Cafeteria Dataset dataset includes 19,632 bite gestures and 3,510 drink gestures, which correspond to 12 hours and 28 minutes of bite, and 5 hours and 13 minutes of drink. Non-intake parts of videos add up to 89 hours and 21 minutes.

3.3.3 Video Processing

To minimize the irrelevant backgrounds and reduce the computation when training models, videos are cropped spatially. Cropping was done by manually selecting a rectangular area in each video. Corresponding to the seven camera positions during the data collection, only 7 different cropping rectangles were needed for the entire dataset. Figure 3.3 shows some examples of cropped frames used for our experiments.

For each video, we only used the portion between the first and the last gestures labeled in the original gesture ground truth (including five classes: bite, drink, rest, utensiling and others), so that data captured before and after the meal are ignored.

3.4 Baseline Models

In this section, we introduce three candidate model architectures, as well as the data pre-processing and post-processing methods we used. Finally, we describe metrics used to evaluate the

classifiers.

3.4.1 Model Architecture

Rouast et al. were the only previous group that investigated end-to-end neural networks on video-based intake gesture detection [75]. Their work focused on bite detection and concluded that CNN-LSTM and SlowFast [27] achieved the top performance with F1 scores of 83.6% and 85.8% respectively. We therefore adapt the two models as baseline models on our dataset. We also looked for more supplement to our baseline model zoos in generic video classification field. We then decide to adapt the large-scale X3D (X3D-L) model [26], which achieved the state-of-art performance by date with top-1 and top-5 accuracies of 92.9% and 77.5% respectively on Kinetics-400 dataset [52], a 400-class large-scale video dataset. Here are the implementation details of the three baseline models.

3.4.1.1 CNN-LSTM

The CNN-LSTM architecture adds a RNN architecture on the top of a 2D-CNN architecture. The 2D-CNN part learns spatial information of every frame and the RNN part learns temporal information across multiple frames. The model input is a sequence of frames, an the output is a set of 3 probabilities ("bite", "drink", "non-intake" classes) for each frame.

We follow Rouast et al. work to implement the model. Specifically, ResNet-50 [40] is used as the 2D-CNN architecture, and a LSTM layer with 128 units is used as the RNN architecture. The input for the model is a 2-second frame sequence sampled at 8 Hz, and each frame is resized to 224 by 224. Every frame is accounted for in loss calculation during the training, and only the prediction for the last frame of every input sequence is used during the testing.

3.4.1.2 SlowFast

The SlowFast architecture uses 3D-CNN layers to travel across frames with two different temporal speeds. The slow pathway is with a low frame rate (i.e. large stride) and focuses on capturing semantic information of a few frames, while the fast pathway is with a high frame rate (i.e. small stride) and focuses on capturing rapidly changing motions. The two pathways are fused by several lateral connections.

Because of some ambiguity in the implementation detail provided by Rouast et al., we follows the original SlowFast proposer [27] to implement the model. The 3D ResNet-50 architecture

is used as the backbones for both pathways. The input duration is 2 second for both pathways. and the frame rates for the fast and slow pathways are 16Hz and 4 Hz respectively. Each frame is resized to 224 by 224. The model output is one set of probabilities (corresponds to "bite", "drink", "non-intake" classes in our case) for one input and is used as the prediction for the last frame in the input.

3.4.1.3 X3D-L

X3D is a family of efficient video networks. X3D optimized the model parameters of 3D ResNet architecture, such as frame rate, input duration, frame size, the number of 3D blocks, kernel sizes and filter sizes. The optimized parameters enable X3D to achieve the best accuracy within a restrained model complexity. We implement the large-scale X3D model, namely X3D-L, as one of our baseline model.

We implement the X3D-L model with the optimized model parameters provided by the original X3D work [26]. The input for the model is a 16-frame sequence sampled at 6 Hz, and each frame is resized to 312 by 312. Same as SlowFast model, we consider the model output as the prediction for the last frame in the input.

3.4.2 Data Processing

Frames are extracted from cropped videos and resized with specified sampling rate and frame size for each model. Afterward, frames' pixel values are processed by z-score standardization.

A sliding window method is utilized to create input sequences and label sequences for the neural network model. The sliding stride is half of the sliding window size on the train and validation set to avoid highly similar samples. The sliding stride is 1 frame at the sampling rate of the model input during the test set so that the output predictions cover every target frame in test videos.

All 264 participants are randomly allocated for train set, validation set and test set. Train set includes 70% participants (i.e. 184 participants), and each of validation and test set includes 15% participants (i.e. 40 participants).

3.4.3 Training

Frames in train set are augmented by dynamically and randomly horizontally flipping and adding jitters to brightness. For each model, a dropout layer with 0.5 dropout rate is used before the last dense layer to help reduce overfitting.

The categorical cross-entropy loss is used to train the neural network. Because of an imbalance between the three classes, class weights are applied to the cross-entropy loss when calculating batch loss. Class weights are calculated on train set by Equation 3.1.

$$w_i = \frac{a}{\sqrt{N(i)}} \tag{3.1}$$

Where w_i and $N(i)$ are the weight and sample numbers for class i, and a is set to 10,000 to avoid very small weights.

The ResNet-50 part of CNN-LSTM is initialized using weights pre-trained on ImageNet [19] dataset. SlowFast and X3D-L are initialized using weights pre-trained on Kinetics-400 dataset. Other model weights are randomly initialized.

The Adam optimizer [54] with an initial learning rate of 0.0001 is used. The learning rate decays exponentially during training. The training runs for 50 epochs.

The training had already converged before 50 epochs when we monitored the training loss. The batch size is set to 8 and each epoch include 10,000 steps. The training process was conducted on two V100 GPUs. The best model of all training epochs is selected according to the unweighted average recall (UAR) over all classes calculated on validation set, as done by Rouast et al. [75].

3.4.4 Post Processing

Using a sliding window approach on the test set, each individual frame gets classified. A smoothing filter is then applied on video-length gesture-level predictions to reduce noise and jitters. The filter includes two parts, filling small gaps between two consecutive intake gestures with the same class, and removing extremely short intake gestures. The thresholds for determining a small gap and a short gesture are 0.5 second for both bite and drink gestures. The thresholds are short so that most predictions for natural intake gestures are preserved.

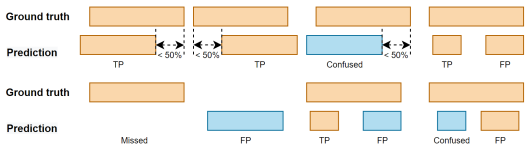


Figure 3.4: Different cases when matching predicted gestures with ground truth. Blocks in yellow and blue stand for bite and drink gestures respectively.

3.4.5 Evaluation Matrix

We evaluate our accuracy on both frame-by-frame classification and gesture detection.

For frame classification, the predicted label of each frame is compared against the ground truth. We calculate recall, precision and F1 score on all three classes (bite, drink and non-intake).

For gesture detection, we evaluate the results for identifying gestures in terms of the gesture count and the alignment on the time span. Our evaluation scheme combines the scheme our group used for measuring the inter-rater reliability during ground truth labeling [87, 88], and the intake gesture counting scheme used by Kyritsis et al. [55] and Rouast et al. [75]. Specifically, a true positive (TP) detection happens when the predicted gesture and corresponding ground truth gesture have the same label and more than 50% overlap. A predicted gesture is considered false positive (FP) if it does not meet the overlap criteria or is not the first detection within the same ground truth gesture event. A confused detection happens when the predicted gesture and corresponding ground truth gesture meet the overlap criteria but are with different labels. A missed detection happens when there is a ground truth gesture but no predicted gesture meeting the overlap criteria. The examples in Fig. 3.4 illustrate those definitions. We calculate precision, recall and F1 score for bite and drink gestures. Note that non-intake cannot be evaluated at the gesture level as it is the background value.

3.5 Results

The frame-level evaluation results reveal the native performance of neural network models. Table 3.2 shows the results on all frames in the test set. It can be seen that X3D-L model achieved the best overall performance on identifying individual bite and drink frames (F1 scores were 0.818 and 0.938 respectively).

Some of the detection mistakes can be explained by boundary errors. The start and end time of each detected gesture do not always align exactly with the ground truth boundaries. Mis-

Table 3.2: Frame-wise classification results per class. The highest numbers are bold.

Model \ Index	Precision		Recall		F1	
	Bite	Drink	Bite	Drink	Bite	Drink
CNN-LSTM	0.723	0.623	0.790	0.894	0.755	0.735
SlowFast	0.756	0.746	**0.851**	0.937	0.800	0.831
X3D-L	**0.798**	**0.756**	0.839	**0.940**	**0.818**	**0.938**

Table 3.3: Gesture detection results per class. The highest numbers are bold.

Model \ Index	Precision		Recall		F1	
	Bite	Drink	Bite	Drink	Bite	Drink
CNN-LSTM	0.857	0.582	0.917	0.787	0.886	0.670
SlowFast	0.833	**0.739**	0.927	0.822	0.877	**0.778**
X3D-L	**0.858**	0.701	**0.943**	**0.849**	**0.899**	0.768

Table 3.4: Confusion matrix for X3D-L model gesture detection.

Actual \ Predicted	Bite	Drink	Non-intake	Total actual
Bite	2,850	2	173	3,025
Drink	2	467	83	552
Non-intake	471	199	-[1]	-
Total Predicted	3,323	749	-	-

[1] "non-intake" gesture class is the continuous background and counting it with the unit of "gesture" is meaningless.

alignments of their boundaries can thus easily contribute to 10-20% error in individual frame classifications. On the other hand, small boundary errors are tolerable since intake gesture detection does not require accurate localization on frame-level resolution. All three models achieved higher recalls than precisions on both gesture detections, meaning that detection mistakes tended to be false positives. One main reason is that the detection jitters lead to a large amount of false positive detections. Note that a smoothing filter is applied on the gesture-level result and most frame-level detection jitters can be removed. All these factors demonstrate why gesture-level evaluation is more important than frame-level evaluation. The main purpose of our study is to detect and count intake gestures.

Table 3.3 shows the results of our method on identifying intake gestures. The best F1 scores on bite and drink gestures were 0.899 and 0.778, which were obtained by X3D-L and SlowFast, respectively. Furthermore, the indices of drink gestures tended to be lower than those of bite gestures. Considering that the number of actual drink gestures was far less than the number of bite

gestures, it may be that additional training data is needed. Some subjects did not take any drink gestures during their recordings.

Because X3D-L model achieved the best performance on most evaluation indices, we pick the model for further analysis. Table 3.4 shows the confusion matrix. We can see that there is very little confusion between bite and drink gestures (2 for both gestures), and most of the errors are the result of missing actual intake gestures and false positive detections of intake gestures. The reasons for these failures include the model's limitations, extreme lighting conditions, and brief intake gestures that do not have enough frame data for video analysis. Another reason for missing gestures is that some appear "unfinished". For example, a subject may start a bite gesture but pause midway to talk with another diner. We found that start-up phases of unfinished bite gestures tended to be classified as bite gestures.

3.6 Conclusion

In this study, we addressed the lack of a video dataset for intake gesture detection in a realistic scenario. We collected a large amount of video data in a real restaurant with little restrictions placed upon participants. We built three baseline models using CNN-LSTM, SlowFast and X3D architectures. The best F1 scores are 0.899 and 0.778 on detecting bite and drink gestures respectively.

Some previous studies can serve as references for our baseline model. Luktuke et al. recognized gestures using the wrist motion data from the Clemson Cafeteria Dataset [58], and achieved average true positive rates (i.e., recall) per participant of 0.797 and 0.847 on bite and drink gestures, respectively.

Our overall results are close to these numbers. This shows that recognizing intake gestures can be done from video about as well as it can be done from wrist motion.

However, there remain some limitations on our dataset. For example, our dataset is limited to one restaurant scenario, which potentially restrains the ability for training general models. Besides, we notice that our trained models were not as generalized on drink gestures as on bite gestures, because the drink gestures were much less than bite gestures. This will guide us in possible future dataset expansion and data augmentation research.

Some advanced post-processing methods can be adapted to process frame-level predictions.

For example, instead of blindly picking the class with the highest probability as the predicted label for a frame, a Gaussian Mixture Model (GMM) can be used to precisely optimize the rule for determining predicted classes of frames. This way refines a post-processing rule from a large amount of frame samples and therefore improves the overall classification performance.

Combining multiple sensor modalities can be another possible approach for improving intake gesture detection performance. For example, our Cafeteria dataset includes data from the wrist sensor, camera, and scale. We expect that more sensor data besides videos introduce more information, and therefore benefit the intake gesture detection.

All of these ideas are topics for future work.

Chapter 4

Video-based Intake Gesture Recognition using Meal-length Context

4.1 Methods

In this section, we first overview the proposed pipeline and the dataset. Then we describe the details including local encoders, global detectors, augmentation method, and the training, testing and evaluation of the proposed pipeline.

4.1.1 Overview

The input to our complete pipeline is a meal-length video, and the output is a label of each frame from the set of classes {bite, drink, non-intake}. Fig. 4.1 shows the training phase of our proposed method. We first build *local encoders* by adapting small-scale versions of SOTA window-based video action recognition networks. During training, local encoders make frame-wise predictions via a sliding window on videos. One local encoder network is trained ten times to obtain ten variants. Then we use frame-wise unnormalized probabilities obtained by local encoders to construct meal-length feature patterns, which we refer to as *global patterns*. During the construction, we augment the

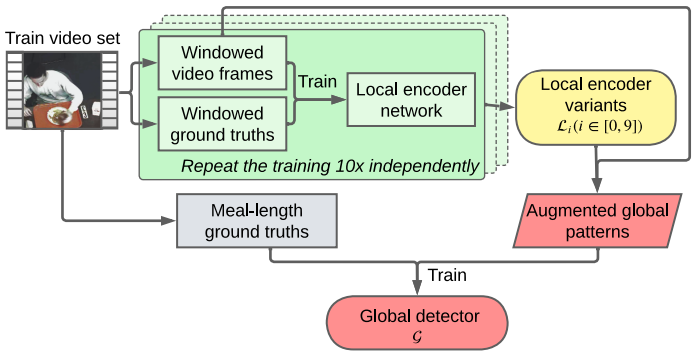


Figure 4.1: Proposed training procedure. Rounded boxes specify the output from the training phase. Window-level operations are represented in green, trained local encoders are in yellow, and meal-level operations are in red.

global pattern dataset by 160x by leveraging model volatility. Finally, with the augmented global pattern dataset, we are able to train a second network which we refer to as the *global detector*. The global detector learns global knowledge, such as the interaction between different gestures and temporal gesture distributions. Therefore, the global detector is expected to perform better than window-based networks.

Fig. 4.2 shows the testing and inference phase of our proposed method. During this phase, a prediction pipeline is established by combining one trained local encoder variant with one trained global detector. The trained local encoder variant constructs a global pattern for each video, and the trained global detector then analyzes these global patterns and predicts labels for frames in each corresponding meal video.

Finally, the predicted labels are evaluated by comparing them to the corresponding ground truth.

4.1.2 Dataset

In this study we used the publicly available Clemson Cafeteria dataset[1] [97] and EatSense dataset [2] [74]. The Clemson Cafeteria dataset includes 486 videos of 264 participants, in which each participant ate a single meal consisting of 1-4 courses (each course was recorded as a separate video) at the Harcombe Dining Hall at Clemson University. Each course is considered an independent meal episode in our proposed pipeline. The participant group consists of 137 females and 127 males, of

[1] `https://cecas.clemson.edu/ahoover/cafeteria/`
[2] `https://groups.inf.ed.ac.uk/vision/DATASETS/EATSENSE/`

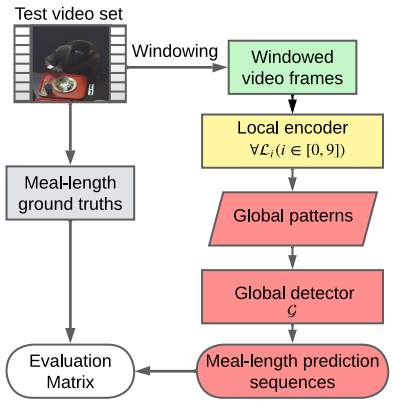


Figure 4.2: Proposed inference and testing procedure. Only one trained local encoder variant is deployed during this phase. Rounded boxes specify the output from the testing and inference phase.

which 183 were of 18-30 years old, 60 were 31-50 years old, and 21 were between 51-75 years old when the data was collected. A total of 374 different food and beverage types appeared in their meals. Utensils included forks, spoons, chopsticks and hands. Containers included plates, bowls, glasses and cups [97]. Videos were recorded in 480 x 640 spatial resolution at 30 Hz. These were cropped to regions of interest centered on the participant and their tray of food and beverage [97].

All 264 participants were randomly allocated to train set, validation set and test set. The train set includes 70% of participants (i.e., 184 participants), and each of the validation and test set includes 15% of participants (i.e. 40 and 39 participants, respectively).

The EatSense dataset includes 135 untrimmed videos from 27 participants of 12 different nationalities. The participant group consists of 11 females and 16 males, of which 11 were below 30 years old, 6 were 30-39 years old, 3 were 40-49 years old, 3 were 50-59 years old, 4 were above 60 years old. Utensils included forks, spoons, knifes, and hands. Videos were recorded in 640 x 480 spatial resolution at 15 Hz or 30 Hz. We do not crop videos in the EatSense dataset, as each video features the participant as the sole subject.

Because of the relatively small number of participants, we randomly split the videos into training, validation, and testing sets, rather than splitting by participants, to minimize performance variability due to dataset division. The train set includes 70% of videos (i.e. 94 videos), and the validation and test sets each include 15% of videos (i.e. 20 and 21 videos, respectively).

In this study, we focus on intake gesture detection and consider bite and drink gestures as target classes. All other classes are consolidated into a single background class. Raw video frames from both datasets are resized according to the specifications recommended for each local encoder

and benchmark method. These details are explained in sections 4.1.3 and 4.1.6.

4.1.3 Local Encoder

We investigate two state-of-the-art networks for local encoders, X3D [26] and CNN-LSTM [20]. The CNN-LSTM architecture achieved the best results in intake gesture detection in the study by Rouast et al. [75]. X3D achieved the best accuracy to date on Kinetics [11], a large dataset for generic video action recognition, with top-1 and top-5 class accuracies out of 700 total classes being 80.0% and 94.5%. As intake gesture detection can be viewed as a downstream task in the realm of generic action recognition, we anticipate that X3D can yield comparable SOTA performance in this domain. Prior to training, the ResNet component of the CNN-LSTM model and the complete X3D are pre-trained on ImageNet dataset [19] and Kinetics datasets [11], respectively. This initialization process ensures that the models are equipped with a strong foundation of visual features relevant to the task of intake gesture detection, enabling efficient training on datasets with relatively small amounts of data.

4.1.3.1 Local Encoder Instantiations

We instantiate the local encoders at small scales (X3D-S and CNN-LSTM-S) to decrease computational cost. This is important because the local encoders will be trained repeatedly for generating and augmenting global pattern data. Table 4.1 shows the instantiation details of each local encoder. We follow the model design approach of the original authors to set the window length and sample rate. Therefore, each local encoder analyzes approximately a 2-second window of video data. A sliding window is used to analyze each video in its entirety.

In addition to keeping the model complexity small, the local encoders are modified to output predictions for every temporal frame in a window (i.e. seq2seq, the input and output keep the same time dimension). This increases the number of samples generated per training cycle, which increases the amount of data augmentation that can be done each time the models are retrained. The data augmentation will be further explained in Section 4.1.4. For X3D-S, the last spatiotemporal pooling layer is set to 1 temporal stride so that the time dimension does not shrink. For CNN-LSTM-S, ResNet34 is used as the backbone. ResNet34 reduces the model complexity while still providing competitive performance to ResNet50. The LSTM layer in CNN-LSTM-S is bidirectional, so that the receptive field of each node equally covers the whole input pattern.

Table 4.1: Instantiations for local encoders. Kernel, stride and output sizes are expressed in temporal size × spatial size. Two SOTA models are downsized and adapted for fast feature encoding purposes.

Stage	X3D-S *13 frames at 5 Hz*		CNN-LSTM-S *16 frames at 8 Hz*	
	kernels	output size	kernels	output size
input layer	-	13×160^2	-	16×224^2
conv_1	1×3^2, 24 channels; stride 1×2^2	13×80^2	7^2, 64 channels; stride 1×2^2	16×112^2
pool_1	-	-	3^2; stride 1×2^2	16×56^2
res_2	$\begin{bmatrix} 1 \times 1^2, 54 \\ 3 \times 3^2, 54 \\ 1 \times 1^2, 24 \end{bmatrix} \times 3$	13×40^2	$\begin{bmatrix} 1^2, 64 \\ 3^2, 64 \end{bmatrix} \times 3$	16×56^2
res_3	$\begin{bmatrix} 1 \times 1^2, 108 \\ 3 \times 3^2, 108 \\ 1 \times 1^2, 48 \end{bmatrix} \times 5$	13×20^2	$\begin{bmatrix} 1^2, 128 \\ 3^2, 128 \end{bmatrix} \times 4$	16×28^2
res_4	$\begin{bmatrix} 1 \times 1^2, 216 \\ 3 \times 3^2, 216 \\ 1 \times 1^2, 96 \end{bmatrix} \times 11$	13×10^2	$\begin{bmatrix} 1^2, 256 \\ 3^2, 256 \end{bmatrix} \times 6$	16×14^2
res_5	$\begin{bmatrix} 1 \times 1^2, 432 \\ 3 \times 3^2, 432 \\ 1 \times 1^2, 192 \end{bmatrix} \times 7$	13×5^2	$\begin{bmatrix} 1^2, 512 \\ 3^2, 512 \end{bmatrix} \times 3$	16×7^2
conv_2	1×1^2, 432 channels; stride 1×1^2	13×5^2	-	-
pool_2	1×5^2; stride 1×1^2	13×1^2	1×7^2; stride 1×1^2	16×1^2
flatten	-	13×1	-	16×1
lstm	-	-	128 units, bidirectional	16×1
dense	3 nodes per timestep	13×3	3 nodes per timestep	16×3

4.1.3.2 Construct Global Patterns

We utilize the 3-class unnormalized probability set, generated by the last dense layer of the local encoder, as the encoded feature vector for each frame. Subsequently, we construct a meal-length global pattern by concatenating the 3-class unnormalized probability sets of each frame in a meal video, where each class (bite, drink, non-intake) corresponds to one channel of global patterns. Using a probability value for each class per frame allows global patterns to explicitly convey gesture information encoded by window-based models in a compact way. Additionally, a 3-class unnormalized probability set is not squashed by the last softmax activation function in a local encoder, making it more continuous and providing more differentiable information about gesture prediction confidence compared to a model's output probabilities that tend to discretely distribute near 0 or 1.

Fig. 4.3 shows the process of constructing meal-length global patterns. When applying a local encoder to a window of video frames, each frame gets a 3-class unnormalized probability set. When sliding a window across a video with the stride of 1, a frame offset can be chosen to record a sequence of unnormalized probabilities, which becomes one global pattern for the video. Different

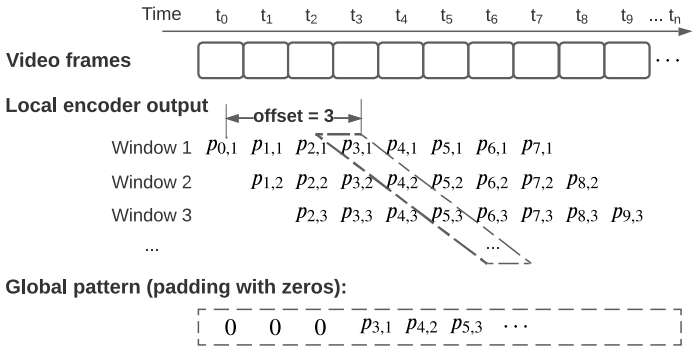


Figure 4.3: An example of constructing global patterns using local encoder outputs. The window size is 8 frames, and the frame offset is 3. $p_{i,j}$ is the unnormalized probability set for the i-th frame in the video, and the frame is fed into the local encoder within the j-th window. The dashed box highlights the unnormalized probabilities taken for constructing global patterns. Those uncovered frames due to the offset are padded with zeros.

global patterns can be generated from one video using different frame offsets; this is discussed more in the next section.

4.1.4 Global Pattern Augmentation

Model volatility in local encoders enables the generation of global patterns across various contexts. We augment global patterns by a total of 130x to 160x using two different sources of model volatility.

4.1.4.1 Using Different Frame Offsets in the Sliding Window

Because local encoders are built for frame-wise predictions, the model outputs for each frame in each input window are accessible. By using different frame offsets within the sliding window, multiple global patterns can be constructed from one video depending on the window size.

Fig. 4.4 shows an example. With a window size of 16 frames, global patterns can be augmented to 16x the number of videos.

When creating a global sample, the window offset remains consistent across all local windows, so the temporal distance for looking forward and backward within a local window is uniform and the distance between data points is consistent.

Different frame offsets affect the distance a local model considers when looking ahead and backward within a window, altering the information used for inferring a frame. Consequently, a local model can yield varying performance across different window offsets, and result in diverse

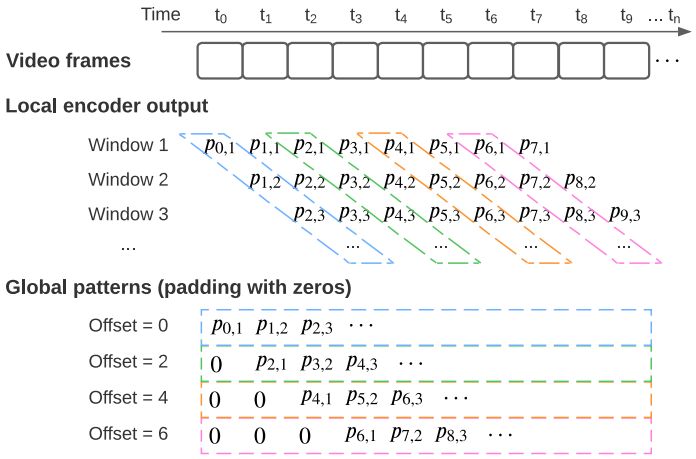


Figure 4.4: An example of generating global patterns using different frame offsets in the sliding window. The window size is 8 frames, and the frame offsets are 0, 2, 4, 6. $p_{i,j}$ is the unnormalized probability set for the i-th frame in the video, and the frame is fed into the local encoder within the j-th window. The dashed boxes highlight the unnormalized probabilities taken for constructing global patterns. Different colors of those boxes stand for different offsets. Those uncovered frames due to the offsets are padded with zeros.

global samples, even when encoding the same raw recording. We regard this variability as a type of model volatility that helps produce more patterns from a limited dataset.

Fig. 4.5 shows an example of the fluctuation range in those global patterns constructed using one local encoder and different frame offsets on one video. We can see that the probability fluctuation between global patterns is apparent. For example, the upper bound of bite at around 930 s indicates a suspected bite gesture with considerable unnormalized probability, while the lower bound does not. Consequently, some global patterns include the suspected bite gesture while others do not. Those diverse contexts augment the global pattern dataset.

Furthermore, we observed that suspicious gestures are often where the local encoder struggles, such as when a subject plays with a utensil or the video has significant noise. A global detector trained on the augmented global pattern dataset is expected to perform better on those places after learning the knowledge of gesture distribution over a meal.

4.1.4.2 Using Separately Trained Local Encoder Variants

We noticed that different local encoder variants of the same network architecture but trained separately do not always make identical inferences. For each of the local encoder instantiations, we train 10 times independently to obtain 10 trained variants. Each trained variant is then used for

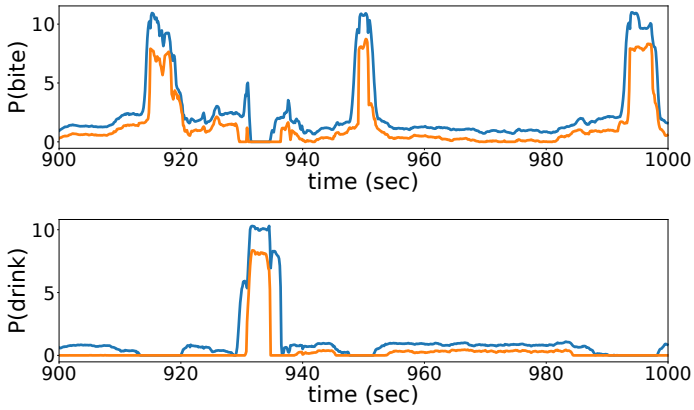


Figure 4.5: An example of the model volatility between different frame offsets. Y axes are the probabilities of bite and drink in the top and bottom plots, separately. The two curves with each plot are upper and lower bounds in global patterns constructed using different frame offsets on one video. 16 global patterns are generated from one trained CNN-LSTM-S variant and 16 different frame offsets. The shown period is from 900 s to 1,000 s in p110/c1 video in Cafeteria dataset.

generating global patterns. This augments the global patterns by an additional 10x.

Fig. 4.6 shows an example of the fluctuation range in those global patterns constructed using separately trained local encoder variants. We can see the obvious gap between the upper and lower bounds, which indicates a significant difference between global patterns in terms of the probability ranges. For example, the upper bound for the bite probability curve indicates a suspected bite gesture with considerable unnormalized probability right after the first detected bite gesture, while the lower bound does not. Moreover, the upper bound for the drink probability curve indicates a second suspected drink gesture with an impulse of the unnormalized probability at around 940 s, while the lower bound does not. These observed differences demonstrate that augmented global patterns get diverse on both the frame-level probabilities and gesture distribution.

One reason that independently trained local encoders get different performance is because of random elements during the training, such as network parameter initialization, the sequence of feeding training patterns, and some random choices within the gradient descent. Such volatility has a random impact when a trained model makes a new inference, and can be considered a realistic and random factor in the augmentation.

The augmented train set contains 53,280 global patterns (160x the video count) when using CNN-LSTM-S as the local encoder (window: 16 frames at 8 Hz), or 43,290 global patterns (130x the video count) when using X3D-S as the local encoder (window: 13 frames at 6 Hz). Therefore, the augmented train set provides sufficient data for training a neural network for meal-length level

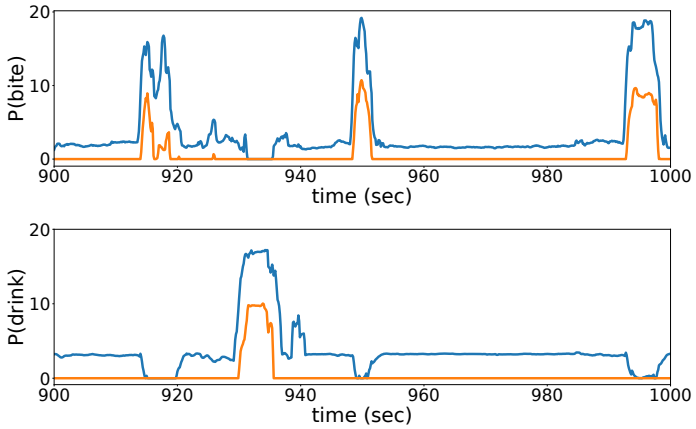


Figure 4.6: An example of the model volatility between independently trained local encoder variants. Y axes are the probabilities of bite and drink in the top and bottom plots, separately. The two curves within each plot are upper and lower bounds of global patterns constructed using one video and separately trained local encoder variants. 10 global patterns are generated by 10 independently trained CNN-LSTM-S variants. The window is 16 frames at 8 Hz, and the frame offset is 7. The shown period is from 900 s to 1,000 s in p110/c1 video in Cafeteria dataset (same as Fig. 4.5).

gesture prediction.

4.1.5 Global Detector

4.1.5.1 Dataset Split and Usage

Data leakage is prevented by performing all data augmentation on the training and validation sets only. The validation set is held separately and is used for selecting the best epoch during the training of the local encoders. The test set is also held separately and is only used for evaluation after all training has been completed.

4.1.5.2 Pre-processing Global Patterns

All values in global patterns are z-score standardized using means and standard deviations calculated from training sets.

Global patterns are then padded to a uniform length by adding zeros to the end, so that the input to the global detector network stays consistent. Since all samples are normalized using z-score standardization prior to padding, a padded value of zero does not impact the distribution. Additionally, these padded locations are masked out during the calculation of the loss function, so they do not influence the optimization process for parameters associated with the actual data

locations.

We set the target global pattern lengths to slightly exceed the longest video in each dataset: 2,500 seconds (approximately 41 minutes) for the Clemson Cafeteria dataset and 1,050 seconds (approximately 18 minutes) for the EatSense dataset.

Subsequently, global patterns are downsampled to reduce sequence lengths and computational costs. Since local encoders handle frame-level information at higher frame rates and global patterns mainly convey gesture-level information, maintaining a high frame rate for global patterns is unnecessary. For the Clemson Cafeteria dataset, patterns are downsampled to 2 Hz, resulting in 5,000 data points per pattern. For the EatSense dataset, patterns are downsampled to 4 Hz, yielding 4,200 data points per pattern. The chosen sampling frequencies are specifically selected based on the durations of labeled bite and drink events in each dataset, ensuring that the downsampled data retain sufficient data points to accurately represent the underlying behaviors.

4.1.5.3 Global Detector Architecture

The global detector network can be any structure. For validation purposes, we choose a simple recursive network as the global detector in this work. The network consists of one bidirectional LSTM layer and one dense layer. The input is the global patterns with 2 Hz and 5,000 data points, and the output is the predicted labels for every data point.

4.1.6 Benchmark Methods

For comparative evaluation, we implemented three SOTA networks: SlowFast [27], X3D-L [26], and CNN-LSTM [20]. SlowFast uses two pathways: a slow pathway at a low frame rate to capture spatial semantics, and a fast pathway at a high frame rate to capture motion at fine temporal resolution. The two pathways are fused by several lateral connections. By treating the raw video at different temporal rates, the method allows the two pathways to have their own expertise on video modeling. X3D is a family of 3D-CNN networks optimized for different computation budgets. Given a specific target model complexity, a X3D variant is constructed by expanding a tiny 2D-CNN architecture along axes of space, time, width and depth with a step-by-step optimization. CNN-LSTM adds LSTMs on top of the 2D-CNN architecture and learns spatial and temporal features sequentially. Though CNN-LSTM cannot learn low-level spatiotemporal features, the architecture leverages LSTMs to better learn long-term dependencies than 3D-CNNs.

Table 4.2: Instantiations for benchmarking. Kernel, stride and output sizes are expressed in temporal size × spatial size

Stage	SlowFast [27] *slow: 32 frames at 16 Hz, fast: 8 frames at 4 Hz*			X3D-L [26] *16 frames at 6 Hz*		Rouast et al. CNN-LSTM [75] *16 frames at 8 Hz*	
	kernels (*slow*)	kernels (*fast*)	output size	kernels	output size	kernels	output size
input layer	-	-	*slow*: 8×224^2 *fast*: 32×224^2	-	16×312^2	-	16×224^2
conv_1	1×7^2, 64 channels stride 1×2^2	5×7^2, 8 channels stride 1×2^2	*slow*: 8×112^2 *fast*: 32×112^2	1×3^2, 24 channels stride 1×2^2	16×156^2	7^2, 64 channels stride 1×2^2	16×112^2
pool_1	1×3^2 stride 1×2^2	1×3^2 stride 1×2^2	*slow*: 8×56^2 *fast*: 32×56^2	-	-	3^2 stride 1×2^2	16×56^2
res_2	$\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 8 \\ 1 \times 3^2, 8 \\ 1 \times 1^2, 32 \end{bmatrix} \times 3$	*slow*: 8×56^2 *fast*: 32×56^2	$\begin{bmatrix} 1 \times 1^2, 54 \\ 3 \times 3^2, 54 \\ 1 \times 1^2, 24 \end{bmatrix} \times 5$	16×78^2	$\begin{bmatrix} 1^2, 64 \\ 3^2, 64 \\ 1^2, 256 \end{bmatrix} \times 3$	16×56^2
res_3	$\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 1^2, 16 \\ 1 \times 3^2, 16 \\ 1 \times 1^2, 64 \end{bmatrix} \times 4$	*slow*: 8×28^2 *fast*: 32×28^2	$\begin{bmatrix} 1 \times 1^2, 108 \\ 3 \times 3^2, 108 \\ 1 \times 1^2, 48 \end{bmatrix} \times 10$	16×39^2	$\begin{bmatrix} 1^2, 128 \\ 3^2, 128 \\ 1^2, 512 \end{bmatrix} \times 4$	16×28^2
res_4	$\begin{bmatrix} 3 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 1^2, 32 \\ 1 \times 3^2, 32 \\ 1 \times 1^2, 128 \end{bmatrix} \times 6$	*slow*: 8×14^2 *fast*: 32×14^2	$\begin{bmatrix} 1 \times 1^2, 216 \\ 3 \times 3^2, 216 \\ 1 \times 1^2, 96 \end{bmatrix} \times 25$	16×20^2	$\begin{bmatrix} 1^2, 256 \\ 3^2, 256 \\ 1^2, 1024 \end{bmatrix} \times 6$	16×14^2
res_5	$\begin{bmatrix} 3 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	*slow*: 8×7^2 *fast*: 32×7^2	$\begin{bmatrix} 1 \times 1^2, 432 \\ 3 \times 3^2, 432 \\ 1 \times 1^2, 192 \end{bmatrix} \times 15$	16×10^2	$\begin{bmatrix} 1^2, 512 \\ 3^2, 512 \\ 1^2, 2048 \end{bmatrix} \times 3$	16×7^2
conv_2	-	-	-	1×1^2, 432 channels stride 1×1^2	16×5^2	-	-
pool_2	8×7^2 stride 1×1^2	32×7^2 stride 1×1^2	*slow*: 1×1^2 *fast*: 1×1^2	16×10^2 stride 1×1^2	1×1^2	1×7^2 stride 1×1^2	16×1^2
conc	-		1×1^2	-	-	-	-
flatten	-		1×1	-	1×1	-	16×1
lstm	-		-	-	-	128 units	16×1
dense	3 classes		3 classes	3×1	3 classes	3 classes per timestep	16×3 classes

Unlike the small-scale instantiations X3D-S and CNN-LSTM-S that were described previously and used for local encoders, these benchmark methods were implemented using large-scale instantiations to achieve the maximum performance possible by each method. Table 4.2 shows details of the benchmarking instantiations. SlowFast used ResNet50 as the backbone. The speed ratio between the fast and slow pathways is set to 4. We follow the original paper [27] to build lateral connections and non-local blocks. X3D-L differs from X3D-S in that it has deeper and wider structures and more input frames, all of which are optimized for achieving the best accuracy with the larger available model complexity. Rouast et al. [75] built a ResNet50 backboned CNN-LSTM model for eating gesture detection tasks, and achieved good performance. We re-implement their model for result benchmarking.

4.1.7 Training

4.1.7.1 Model Initialization

All window-based models in this study leverage transfer learning, while global detector models are trained from scratch.

Specifically, X3D families and SlowFast are initialized using corresponding models trained on Kinetics [11], a large dataset for video action recognition. ResNet backbones of CNN-LSTM

families are initialized using ResNet34 or ResNet50 trained on the ImageNet dataset [19]. LSTM parts in CNN-LSTM families and global detector models are initialized using weights with random normal distribution.

4.1.7.2 Training Settings

We use the Adam optimizer to train models. The learning rate for global detector models is 0.001, with an exponentially decaying rate of 0.9. Other models are initialized via transfer learning, and the learning rate is 0.0001, with an exponentially decaying rate of 0.9.

The batch size is 32 for global detector models and 8 for all window-based models. Each model is trained for 50 epochs, and the maximum steps within each epoch are 10,000. Afterward, inspired by Rouast et al. [75], the unweighted average recall (UAR) on the validation set is used to decide the best epochs.

The categorical cross-entropy loss is used to train models. Class weights are applied to the cross-entropy loss when calculating batch loss. Class weights are calculated on the training set as:

$$w_i = \frac{a}{\sqrt{N(i)}} \tag{4.1}$$

where w_i and $N(i)$ are the weight and sample numbers for class i, and a is set to 10,000 to avoid very small weights.

To mitigate overfitting, we apply batch normalization [50] after every convolution layer and LSTM layer. Additionally, dropout with a rate of 0.5 is applied before the last dense layer of each model.

The video windows used to train the local encoder are augmented using standard brightness jittering and random horizontal flipping [75].

We trained our models using two Nvidia V100 GPUs. Table 4.3 shows the time spent on one training phase for each model. Local encoders X3D-S and CNN-LSTM-S trained much faster than full-size video action recognition networks.

4.1.8 Inference and Testing

During the testing phase, we integrate one trained local encoder variant with a trained global detector into a pipeline that delivers meal-length predictions from videos. Local encoders are

Table 4.3: Training time for models. Reported hours are for training a single model instance.

Models	Training time (hrs)
X3D-S[a]	24
CNN-LSTM-S[a]	18
Global detector	5
SlowFast	80
X3D-L	118
Rouast et al. CNN-LSTM	30

[a] These models are scaled down and the training of multiple variants can occur in parallel with sufficient computational resources.

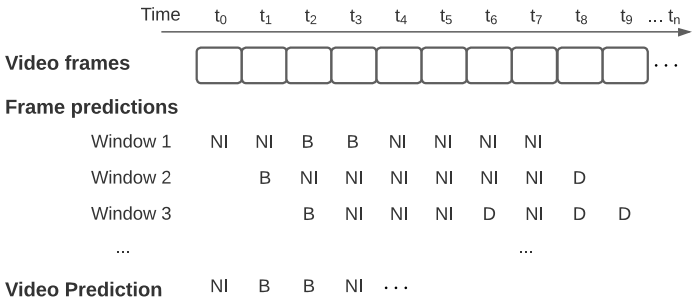


Figure 4.7: An example of constructing meal-length frame predictions using local encoder predictions. The window size is 8 frames. B, D, NI stands for bite, drink and non-intake classes, respectively.

designed to output logits for every frame within an input window, primarily to generate more global patterns. This sequence-to-sequence prediction mechanism results in each frame being classified multiple times as it appears within different windows due to the sliding window approach. Unlike benchmark models, which operate on a sequence-to-one basis and make a single classification prediction per input window, this can introduce variability. To mitigate fluctuation caused by different window offsets, we implement a maximum vote strategy as a post-processing step. This strategy fuses outputs from different window offsets, as shown in Figure 4.7. In cases of a tie among three classes, we prioritize classification as bite, drink, and non-intake, in that order, to emphasize intake detection.

Benchmark models are direct instantiations from the original papers and make one classification prediction for an input window. Therefore, we use these models to predict the last frame of each window. After the sliding window travels through one video with a stride of 1, the predictions form a frame-wise prediction sequence for the video.

For a comprehensive evaluation of the global detector performance, we independently test all potential combinations of local encoder variants with the trained global detector, and report

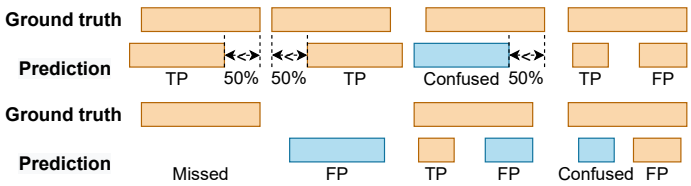


Figure 4.8: Different cases when matching predicted gestures with ground truth. Blocks in yellow and blue stand for bite and drink gestures respectively.

performance distributions including means and standard deviations. Additionally, the local encoder variant that yields the best validation results is selected for combination with the global detector when benchmarking against other studies.

Meal-length frame-level predictions are converted to gesture-level predictions by searching for the start and end times of each detected gesture. We apply a smoothing filter on gesture-level predictions to reduce noise and jitter. The filter fills small gaps between two consecutive intake gestures with the same class, and removes extremely short intake gestures. The thresholds for determining a small gap and a short gesture are both 0.5 second, which is unlikely to filter out natural intake gestures.

4.1.9 Evaluation Metrics

The meal-length prediction of a model is compared with the corresponding ground truth to calculate the model performance. We evaluate model performance on both gesture classification and localization on the time span.

Our evaluation scheme combines the scheme our group used for measuring the inter-rater reliability during ground truth labeling [88], and the intake gesture counting scheme used by Kyritsis et al. [55] and Rouast et al. [75]. Specifically, when a predicted gesture overlaps a ground truth gesture with the same label, if the predicted gesture is the first detection within the range of the ground truth gesture and the overlap ratio is more than 50%, the predicted gesture is considered true positive (TP). Otherwise, the predicted gesture is considered false positive (FP). When the predicted gesture and corresponding ground truth gesture meet the overlap criteria but are with different labels, the predicted gesture is considered as a confused detection. A missed detection happens when there is a ground truth gesture but no predicted gesture meeting the overlap criteria. The examples in Fig. 4.8 illustrate those definitions. We calculate precision, recall and F1 score for

Table 4.4: Model performance on Clemson Cafeteria dataset with and without the proposed framework. Reported F1, precision and recall values are averages from independently testing each of ten trained local encoder variants. Applying our framework on a window-based backbone resulted in significant improvements in F1 scores and most other performance metrics.

(a) Backbone: CNN-LSTM-S

Method	F1		Precision		Recall	
	Bite	Drink	Bite	Drink	Bite	Drink
Window-based	0.89	0.75	0.90	0.73	0.88	0.78
Proposed framework	**0.91**	**0.84**	0.91	0.89	0.91	0.80

(b) Backbone: X3D-S

Method	F1		Precision		Recall	
	Bite	Drink	Bite	Drink	Bite	Drink
Window-based	0.91	0.73	0.88	0.68	0.93	0.80
Proposed framework	**0.94**	**0.85**	0.95	0.91	0.92	0.80

bite and drink gestures. Note that non-intake cannot be evaluated at the gesture level as it is the background value.

4.2 Results and Evaluation

We conduct three sets of comparative evaluations. First, we evaluate the performance of each local encoder with and without the use of the proposed framework on two datasets: Clemson Cafeteria and EatSense. This comparison reveals how much global information over a meal can improve gesture detection performance. Second, we compare the performance of our two-stage framework (local encoder and global detector) with other SOTA video action recognition models on Clemson Cafeteria dataset, the largest intake gesture dataset to date. Third, we investigate how our new method performs across 264 participants on Clemson Cafeteria dataset. This result helps us assess the generalizability of models.

4.2.1 Influence of Using Global Detectors

Table 4.4 shows the evaluation metrics on Clemson Cafeteria dataset for using local encoders alone, compared to combining them with a global detector. We can see that adding a global detector led to an increase in the F1 scores for both bite and drink gestures, for both the CNN-LSTM-S and X3D-S local encoders, with the increase ranging from 0.02 to 0.12. This shows that global meal-length pattern analysis provided a consistent improvement in intake gesture detection. Additionally, recalls tended to remain about the same, but precisions were greatly improved, with the largest

Table 4.5: Model performance on the EatSense dataset with and without the proposed framework. Reported F1, precision and recall values are averages from independently testing each of ten trained local encoder variants. Applying our framework on a window-based backbone resulted in significant improvements in F1 scores and most other performance metrics.

(a) Backbone: CNN-LSTM-S

Method	F1		Precision		Recall	
	Bite	Drink	Bite	Drink	Bite	Drink
Window-based	0.81	0.66	0.75	0.55	0.89	0.83
Proposed framework	**0.84**	**0.74**	0.78	0.64	0.92	0.90

(b) Backbone: X3D-S

Method	F1		Precision		Recall	
	Bite	Drink	Bite	Drink	Bite	Drink
Window-based	0.75	0.59	0.67	0.46	0.85	0.83
Proposed framework	**0.78**	**0.73**	0.72	0.63	0.85	0.89

Table 4.6: Model stability on Clemson Cafeteria dataset with and without the proposed framework. Reported standard deviations are calculated from independently testing each of ten trained local encoder variants. A global detector helped reduce fluctuations between different trained window-based variants in most performance matrices (smaller standard deviations), indicating the benefits of stabilizing model performances across different training runs.

(a) Backbone: CNN-LSTM-S

Method	std: F1		std: Precision		std: Recall	
	Bite	Drink	Bite	Drink	Bite	Drink
Window-based	0.04	0.06	0.06	0.09	0.04	0.04
Proposed framework	**0.03**	**0.02**	0.06	0.02	0.01	0.03

(b) Backbone: X3D-S

Method	std: F1		std: Precision		std: Recall	
	Bite	Drink	Bite	Drink	Bite	Drink
Window-based	0.02	0.05	0.02	0.08	0.01	0.03
Proposed framework	**0.01**	**0.02**	0.01	0.03	0.01	0.02

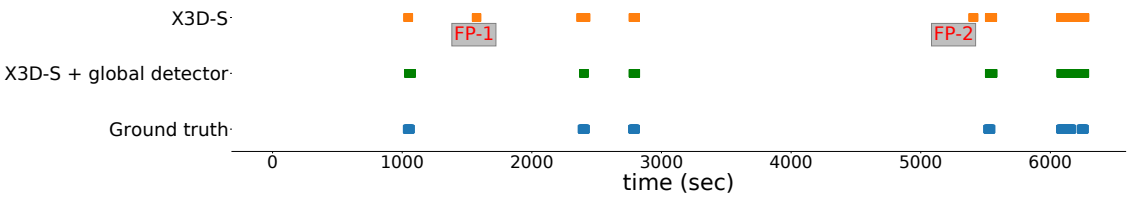
increases occurring for drink gestures (0.16 and 0.23 improvement). This reveals that one main advantage of using global analysis is in reducing false positive detections. The global detector is able to use a longer context to determine if transient detections by the local encoder are actual gestures or not.

Table 4.5 shows the evaluation metrics on the EatSense dataset, comparing the performance of local encoders alone to their combination with a global detector. We can see that F1 scores for both bite and drink gestures increased by 0.03 to 0.14, after integrating a global detector to window-based backbones. Recalls and precisions also improved, especially for the drink class. This provides additional evidence for the effectiveness of our global analysis framework.

When comparing Table 4.4 with Table 4.5, we can see that window-based backbones performed less effective on the EatSense dataset than on the Clemson Cafeteria dataset. A primary factor is the different definitions the two datasets use to define bite and drink classes. The Clemson Cafeteria dataset includes accessory sub-movements, such as moving hands toward the mouth, within the intake event labels. In contrast, the EatSense dataset strictly defines intake events as the brief periods when food or beverage enters the mouth. This labeling results in shorter intake gestures in the EatSense dataset, posing increased detection challenges. Despite these challenges, our global detector consistently enhanced the performance of backbone models across both datasets.

Fig. 4.9 shows an example of how our proposed global detector reduced false positive detections compared to the local encoder on Clemson Cafeteria dataset. The first false positive detection was isolated and did not resemble the appearance of a typical gesture distribution, and the second false positive detection from the local encoder was likely too close to the following detection to be real. In the video, The low light environment and the clothing color similar to the background decreased the visibility of tools (e.g. cups) and hand movements, making it challenging for the model to identify gestures accurately. However, the global detector effectively eliminated false positive detections by analyzing global clues and frame-level probabilities in combination.

Table 4.6 shows the standard deviations of evaluation metrics across ten local encoder variants trained independently with the same model structure, using Clemson Cafeteria dataset. We can see that by wrapping a local encoder into our framework, the standard deviations of most evaluation metrics were reduced. This suggests that our framework effectively stabilized the performance of local encoders between different training phases, making it easier to obtain a well-trained model irrespective of the starting point of training.



(a) Drink gesture prediction results on the video timespan. Two false positive predictions when using X3D-S alone are indicated as 'FP1' and 'FP2'

(b) Clip - FP1: the subject raised the left hand while chatting with others.

(c) Clip - FP2: The subject moved the cup towards their mouth but ultimately did not drink from it.

Figure 4.9: An example of the effectiveness of the global detector in reducing false positive detections of drink gestures. By considering meal-length information and behavior patterns, the global detector successfully eliminated two false positives in the video with index 'p259/c1' from the Clemson Cafeteria dataset.

Table 4.7: Our results compared to SOTA models on Clemson Cafeteria dataset. Our framework utilizing X3D-S as the backbone achieved significantly higher F1 scores while using a much smaller model size during the testing phase.

Model	#Params (M)	F1		Precision		Recall	
		Bite	Drink	Bite	Drink	Bite	Drink
Benchmark: X3D-L	5.34	0.90	0.77	0.86	0.70	0.94	0.85
Benchmark: SlowFast	33,65	0.88	0.78	0.83	0.74	0.93	0.82
Benchmark: Rouast et al. CNN-LSTM	24.62	0.89	0.67	0.86	0.58	0.92	0.79
Ours: CNN-LSTM-S + global detector	21,65[a]	**0.94**	0.86	0.97	0.90	0.92	0.83
Ours: X3D-S + global detector	3.02[a]	0.93	**0.88**	0.95	0.95	0.91	0.81

[a] The reported number of parameters corresponds to one inference pipeline, including one local encoder and one global detector.

4.2.2 Comparison with State-of-the-art Models

We compare our methods with current SOTA networks on Clemson Cafeteria dataset which is the largest intake gesture dataset with untrimmed videos to date. Table 4.7 shows the results. For our results, we used CNN-LSTM-S and X3D-S as local encoders. Each local encoder was trained 10 times for data augmentation, so potentially our pipeline can be evaluated 10 times by combining each variant with our global detector. We report the best pipeline variant according to their performance on the validation set. We can see that the F1 scores of both our framework instances were higher than all the benchmarks for both bite and drink gestures, with the improvement ranging from 0.03-0.06 for bite gestures and 0.08-0.21 for drink gestures. It is thus reasonable to conclude that global patterns over meal episodes benefit gesture detection and improve upon window-based approaches.

The combination of X3D-S and the global detector achieved the highest class-balanced F1 scores and precisions on detecting both bite and drink gestures. X3D-L had higher recalls on detecting both gestures than combined detectors, but at the cost of much lower precisions. Therefore, it can be concluded that X3D-L tended to make more gesture detections, some of which were true positives but more of which were false positives. In general, the combined detector achieved better overall performance than these SOTA networks.

Considering that our framework implementation used downgraded window-based networks as local encoders (small-scale instantiations), it is reasonable to suppose that the performance improvement might be even greater if the local encoders could be trained with large-scale instantiations. This is currently infeasible due to the high computational burden of repeatedly training these models for global pattern data augmentation.

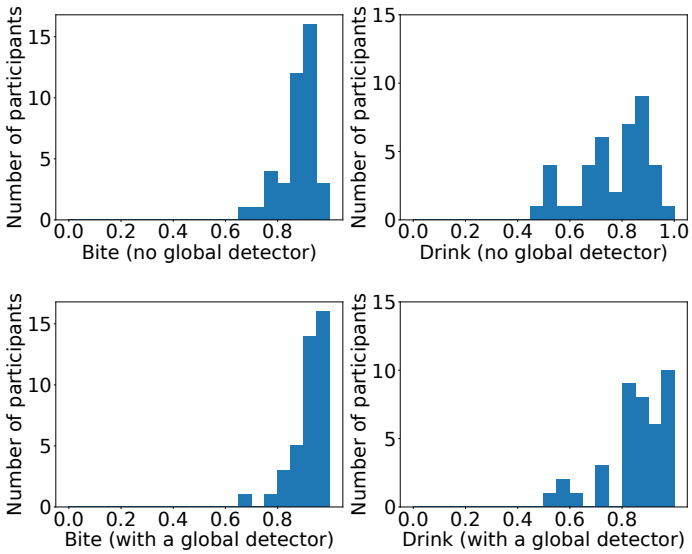


Figure 4.10: F1 score distribution of participants from Clemson Cafeteria dataset using X3D-S alone (bottom row) and combined with a global detector (top row). Our framework with a global detector helped concentrate the subject-wise results in the higher F1 score range, indicating its ability to stabilize model performance across different subjects.

4.2.3 Performance on Different Participants

In this section we analyze the performance improvement between individuals using the Clemson Cafeteria dataset, which is the largest dataset in the field to date. As detailed in Section 4.1.2, the test set was randomly split and includes 39 participants.

Fig. 4.10 shows a histogram of F1 scores per participant from the X3D-S network, with and without integrating the proposed global detector.

It can be observed that after using a global detector, the F1 score distributions concentrated more on higher values. For example, in bite detection, the percentage of participants with F1 scores higher than 0.9 increase from 46% to 74% after utilizing global detectors. In drink detection, the percentage of participants with F1 scores higher than 0.8 increase from 53% to 82% after utilizing global detectors. This suggests that the application of global patterns contributes to consistently improved performance across different individuals. One main reason is that meal videos have general episode-level patterns that a global detector can learn and leverage on inferring other individuals.

We also observe that adding a global detector did not always yield significant F1 score improvement on every participant, and several small bins remained below the high F1 scores mentioned earlier. Therefore, we conducted a demographic analysis to explore the varying performance

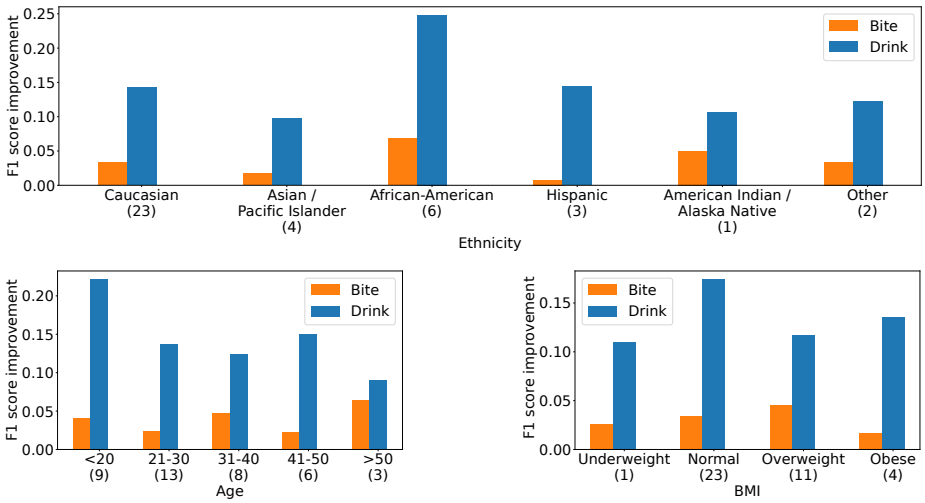


Figure 4.11: F1 score improvements on different sub-populations from Clemson Cafeteria dataset by combining a global detector with one X3D-S local model variant. Numbers in parenthesis are numbers of participants in corresponding categories.

improvements provided by the global detector across different participant groups

Table 4.8 shows the distribution of participants across different demographic categories used for training, validation and testing. We can see that the overall distribution of the testing set closely mirrors that to of the training and validation sets, with small variations observed in minor groups. This indicates that the dataset splitting process did not introduce significant demographic bias during the training of models.

Fig. 4.11 shows the performance improvement of different sub-populations after applying our framework to X3D-S model. Across ethnicities, F1 scores for African American subjects had the largest increases (7% for bite detection, 25% for drink detection). This may be because of the challenge that darker skin color is more likely to blend into the background, which hinders tasks involving recognizing facial expression and hand motions within video. So in this case, the use of global patterns is most helpful. Across all ages, bite gesture F1 scores increased by 2-6%. Drink gesture F1 scores increased more for younger people (22% for subjects younger than 20 years old) than for older people (9% for subjects older than 50 years old). This may be due to a higher consistency in the global distribution of drink intake during a meal (e.g. "washing down foods" towards the end of a meal) for younger people compared to older people. Across all BMI categories, F1 scores for bite and drink gestures increased by 2-5% and 1-7%, respectively. However, there was no clear correlation between BMI groups and F1 score improvement.

Table 4.8: Distribution of participants across demographics in the training, validation, and testing sets split from the Clemson Cafeteria dataset.

Age	¡20	21-30	31-40	41-50	¿50
Train & val	45	131	20	27	18
Test	9	13	8	6	3

BMI[a]	Underweight	Normal	Overweight	Obese
Train & val	4	144	56	35
Test	1	23	11	4

[a] The definition of BMI categories follows WHO standard [65]: Underweight (¡18.5), normal range (18.5-24.9), overweight (25.0-29.9), obese (¿=30.0)

Ethnicity	Caucasian	Asian / Pacific Islander	African-American	Hispanic	American Indian / Alaska Native	Other
Train & val	174	25	20	8	1	13
Test	7	4	6	3	1	2

4.3 Conclusion

This study presented a new approach to detecting intake gestures in videos. We analyzed meal-length patterns that improve the detection of individual gestures. To train this classifier, we described an efficient augmentation method that boosts meal-length patterns by 130x-160x by making use of model volatility. With sufficient meal-length patterns after the augmentation, a second network was trained to model meal-length patterns. Integrating all our ideas, we showed an end-to-end pipeline for intake gesture detection that achieved better overall performance than related SOTA networks, with F1 scores increasing by 0.03-0.06 for bite gestures and 0.08-0.21 for drink gestures.

We hypothesize that the improvement on drink gestures was larger than the improvement on bite gestures because drinks tend to follow a more consistent global pattern between individuals. For example, many people tend to consume more of their beverage towards the end of a meal (the so called "washing down" of food).

In addition to benchmarking SOTA methods, our experiments also investigated the effectiveness of learning global patterns. To make the augmentation phase computationally feasible, we downsized two SOTA networks and evaluated their performance before and after incorporating them into our framework. Our results showed that adding a second stage that considered meal-length patterns improved F1 scores on gesture recognition, particularly for drink gestures. Furthermore, our framework stabilized model performance across different training runs and individuals.

All of these results confirm our hypothesis that individuals share similar behavior patterns

during a meal timespan, which can be learned by our global detector and can improve the accuracy of intake gesture recognition. These findings are consistent with other researchers who have studied long-term video content. For example, Yang et al. [112] consistently obtained over 2% video-level accuracy gain on Kinetics-400 dataset using a collaborative memory pool that captured dependencies across multiple windowed video clips. Similarly, Tang et al. [96] achieved 2.7% better mAP@0.5 on THUMOS'14 dataset by deploying a global attention mechanism on their base model to capture global context from several minutes of video.

An automatic eating activity monitoring system could aid a healthcare professional in delivering guidance and treatment. By deploying a camera within users' homes, such a system can observe their natural eating behaviors in daily life and automatically calculate summaries of their eating patterns, such as eating rate, meal duration, kilocalories consumed [80], and typical drink-to-food ratio. Based on this information, healthcare professionals could provide tailored interventions remotely without imposing heavy recall tasks on users. For example, clinicians can setup real-time triggers into the system to alert users when they spend too much time eating [91] or eat too fast [36]. Furthermore, healthcare consultants can design time-of-day based strategies for users and provide timely adjustments, such as intermittent fasting or adjusting the daily distribution of food intake [5], based on the monitored kilocalories consumed by users.

The insights provided by video data facilitates the development of robust intake gesture detection algorithms, which in turn enables the assessment of other eating activities. By integrating global pattern analysis, our framework demonstrates increased accuracies in recognizing intake gestures on multiple datasets. In addition to achieving SOTA overall accuracies, our framework consistently delivers high performance across participants of diverse demographics. For instance, in testing on the Clemson Cafeteria dataset, 74% of participants achieved F1 scores exceeding 0.9 for bite detection, and 82% exceeded 0.8 for drink detection. This suggests promising potential for real-world deployment in individuals' homes. Additionally, our streamlined inference pipeline features significantly reduced model sizes compared to existing solutions, and therefore require fewer computational resources. This makes it feasible to deploy a user-friendly monitoring system without costly hardware investments.

Our proposed framework could enhance the accuracy of other video-based behavior recognition problems, especially in videos with a consistent theme such as sports or dance. In activities like a basketball game, it is likely that different subjects follow similar routines throughout the activity.

Although the clue from the global view alone is not sufficient to model individual behaviors, it provides useful information about the location of behaviors over the duration of the activity. Analyzing long videos with an end-to-end neural network is computationally expensive, and current researchers can only model videos up to several minutes long. Our framework offers an efficient way to analyze much longer videos while preserving the integrity of the video-length patterns. Our local encoder backbone can be adapted from most SOTA behavior recognition models. And our global detector helps correct many false detections due to noise in transient windows used by those models. Our experimental results demonstrate the effectiveness of a global detector for intake gesture recognition. Since our framework analyzes raw videos without any specific design for meal activity, we expect it to benefit behavior recognition tasks in other activities.

Although our method has shown promising results, it has some limitations. Firstly, our method is not suitable for real-time applications where frames are captured from a continuous stream, as it benefits from analyzing a completed video as one sample. However, this limitation can be bypassed during offline scenarios where videos are recorded and post-processed for accurate behavior logging.

There are several research questions to explore regarding our global pattern augmentation method. Our approach relies on retraining the local encoder multiple times and utilizing different frame offsets to generate sufficient global patterns for training. Each independent execution of training may be considered a new interpretation (model) of what was learned from real data. This idea warrants a much deeper theoretical and experimental evaluation across multiple types of classification problems. There may be limitations in increasing dataset diversity, particularly in scenarios where the dataset is too small or when the local model lacks volatility. Understanding the strengths and limitations of our augmentation method is an important question for future work.

This study found differences in certain demographics in terms of increases in accuracy of gesture recognition from the use of global patterns. Specifically, global patterns benefited African Americans more than other ethnicities, and younger people more than other age groups. However, the dataset was not collected with balanced demographics intended to study these questions, so these analyses are preliminary. Future work could explore these differences more systematically.

There are several other possible improvements to our framework that can be explored in future work. First, instead of using frame-level unnormalized probabilities as the features for modeling meal-length patterns, we could use more preliminary features, such as features from the middle

stages of the encoding network. These would have higher feature dimensions and may convey more information to the global detector. A challenge would be that the global detector would need to be more complex to process the increased input dimensions. Second, the model could be trained for each individual instead of trained on all data. This could allow the classifier to learn personalized meal-length patterns. We saw that a few participants had distinct intake behaviors in the experiment results. Networks trained for these particular participants might improve gesture detection accuracy. However, more data are needed to train personalized networks. All these ideas remain for future work.

Chapter 5

Generalized Framework for Sparse Activity Recognition

This chapter presents a unified framework for recognizing sparse activities in lengthy sensor recordings under limited dataset size. Building on the cross-domain motivation introduced in Section 1.1, we focus on a two-stage pipeline that captures full-recording global context while remaining computationally practical. The chapter details the method, evaluation setup across four modalities (IMU, video, EEG, and audio), and ablation studies that isolate the contribution of global-pattern-level augmentation.

5.1 Methods

5.1.1 Overview

We begin by assuming a dataset consisting of N raw recordings, denoted as X_i^{rec} for $i \in [1, N]$. Each recording is sampled at f_{rec} Hz and has an average duration of t_{rec} hours. Each timestep may contain an arbitrary number of sensor channels or measurements.

Within each recording X_i^{rec}, we define a set of N_i^{event} sparse events, denoted by E_{ij}, where $j \in [1, N_i^{\text{event}}]$. The task is to detect these sparse events E_{ij} across all $i \in [1, N]$ and $j \in [1, N_i^{\text{event}}]$. An event E_{ij} is considered sparse if it satisfies one of the following two conditions:

$$\text{criteria}(E_{ij}) = \begin{cases} \frac{\text{duration}(E_{ij})}{\text{duration}(X_i^{\text{rec}})} < 0.01 & \text{(short duration)} \\ N_i^{\text{event}} < 10 & \text{(low frequency)} \end{cases} \tag{5.1}$$

Additionally, for each $i \in [1, N]$, the total duration of all sparse events must satisfy:

$$\sum_{j=1}^{n_i^{\text{event}}} \frac{\text{duration}(E_{ij})}{\text{duration}(X_i^{\text{rec}})} < 0.6 \tag{5.2}$$

These criteria, derived from the four application domains studied in this work, ensure that the events of interest are not dominant in the recording. Note that the events need not be extremely rare in aggregate, but must follow one of the two sparsity patterns above. Furthermore, the task is not limited to binary classification; other portions of each recording may belong to one or more additional classes. Our primary motivation lies in addressing class imbalance due to sparsity.

We assume the existence of a local detector $\mathcal{L}$ trained to detect sparse events using a window-based approach. The detector $\mathcal{L}$ processes sliding windows of duration t_L, moving with stride S_L over each recording.

For training, a local window dataset $\mathcal{D}_L$ is constructed by segmenting X_i^{rec} into overlapping windows $X_{i,j}^{\text{local}}$, where $j \in [1, N_w]$ and

$$N_w = \frac{\text{duration}(X_i^{\text{rec}}) - \text{duration}(X_{i,j}^{\text{local}})}{S_L} \tag{5.3}$$

Each window is labeled by majority class or per-timestep labels. Since $\text{duration}(X_{i,j})$ and S_L are typically much smaller than $\text{duration}(X_i^{\text{rec}})$, the resulting dataset $\mathcal{D}_L$ of size $N_w \times n$ is usually sufficient to train local detectors of moderate complexity using conventional augmentations such as magnitude perturbations or rotations. Local detectors may operate at a downsampled frequency F_L to reduce computation.

In our framework, the local detector serves as Stage 1. After training, the detector $\mathcal{L}$, with its final decision layer removed, is slid over each raw recording X_i^{rec} to extract a sequence of intermediate feature vectors. Concatenating these features yields a global sample that captures the global pattern of the entire recording. The resulting dataset of global samples is denoted by $\mathcal{D}_G$.

To address data limitations in training the global detector, we use our proposed SCOPE method to expand $\mathcal{D}_G$ by training the local detector M times with different random initializations

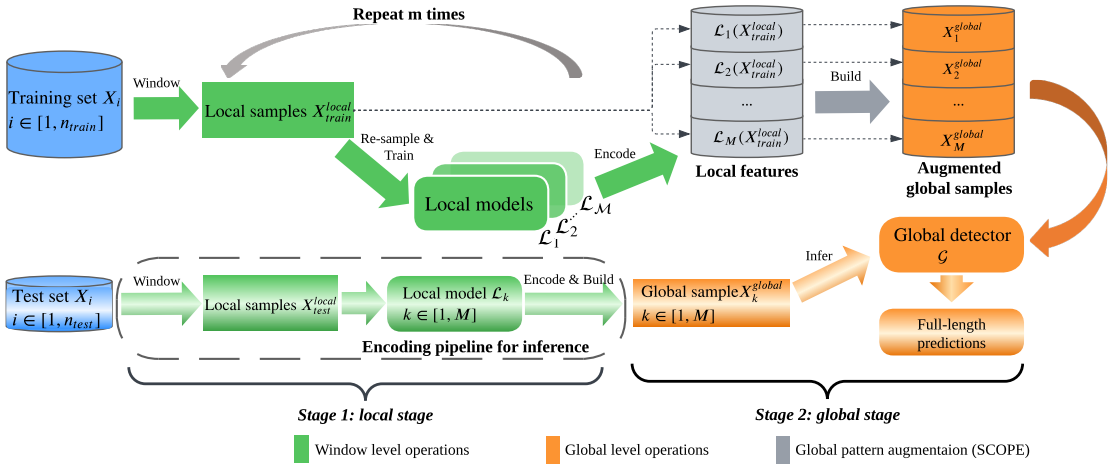


Figure 5.1: Framework Overview. Stage 1 wraps existing window-based models to encode local features. Stage 2 trains a global detector on full-recording samples constructed from local features. SCOPE augmentation generates sufficient global samples after Stage 1.

and re-sampled training subsets. Each training round produces a distinct model $\mathcal{L}_k$, $k \in [1, M]$, which generates a unique global sample for each X_i^{rec}, resulting in $|\mathcal{D}_G| = M \times n$. Depending on the application, we further introduce domain-specific heuristics to increase the size of $\mathcal{D}_G$ by 100–1000× relative to N.

Finally, a global detector $\mathcal{G}$, serving as Stage 2, is trained on these augmented global samples to recognize full-length patterns.

The following subsections detail each stage of the process.

5.1.2 Framework

Our proposed framework, illustrated in Figure 5.1, centers on the application of global detectors for activity recognition. It operates in two distinct stages, each contributing to the overall process of capturing comprehensive global patterns in sensor recordings.

5.1.2.1 Stage 1 – Local Feature Encoding

In the first stage, a window-based model $\mathcal{L}_k$ for $k \in [1, M]$ is trained to capture local temporal patterns in sensor recordings. Given a raw recording X_i^{rec}, we apply a sliding window of duration t_L and stride S_L to obtain overlapping windows $X_{i,j}^{\text{local}}$ for $j \in [1, N_w]$. Then we remove the final prediction layer of $\mathcal{L}$ and run the encoder on these windows to construct an embedding for

the full recording as:

$$X_{i,k}^{\text{global}} = \left[\mathcal{L}_k(X_{i,1}^{\text{local}}), \ldots, \mathcal{L}_k(X_{i,N_w}^{\text{local}})\right] \tag{5.4}$$

This sequence $X_{i,k}^{\text{global}}$ forms a sample in the global dataset $\mathcal{D}_G$.

5.1.2.2 Stage 2 – Global Pattern Learning

The second stage trains a global detector $\mathcal{G}$ on the dataset $\mathcal{D}_G$, where each sample corresponds to a full-length recording encoded by local features. The global detector learns to model long-range dependencies across the full timespan and outputs position-wise predictions.

Formally, the global detector $\mathcal{G}$ is defined as:

$$\mathcal{G} : X_i^{\text{global}} \to \hat{Y}_i = [\hat{y}_{i,1}, \hat{y}_{i,2}, \ldots, \hat{y}_{i,N_w}] \tag{5.5}$$

where each $\hat{y}_{i,j}$ is the predicted class label (or score) for the j-th window in recording i.

To address the limited availability of raw recordings, we introduce a novel data augmentation method named SCOPE (SynthetiC glObal Pattern augmEntation). SCOPE enhances the diversity of global samples by repeating Stage 1 multiple times with different training configurations. Specifically, the process consists of M independent training iterations of the local detector, producing distinct local models $\mathcal{L}_k$ for $k \in [1, M]$.

Each local model $\mathcal{L}_k$ is trained from a randomly initialized state with a randomly resampled subset of the window dataset $\mathcal{D}_L^{(m)}$. For every raw recording X_i^{rec}, this results in M global samples:

$$\begin{aligned} X_{i,k}^{\text{global}} = \left[\mathcal{L}_k(X_{i,1}^{\text{local}}), \ldots, \mathcal{L}_k(X_{i,N_w}^{\text{local}})\right] \\ (i \in [1, N], k \in [1, M]) \end{aligned} \tag{5.6}$$

which collectively expand the size $|\mathcal{D}_G|$ from N to $M \times N$.

Figure 5.2 illustrates the SCOPE procedure for a single iteration k. Each global sample $X_i^{\text{global},(k)}$ is formed by passing a raw recording X_i^{rec} through the encoder $\mathcal{L}_k$ trained with randomization.

After generating the full augmented set of global samples, we train the global detector $\mathcal{G}$ on $\mathcal{D}_G$. During inference, $\mathcal{G}$ processes the entire sequence of local embeddings from a test recording and outputs predictions across all timepoints. The model is expected to learn to integrate features

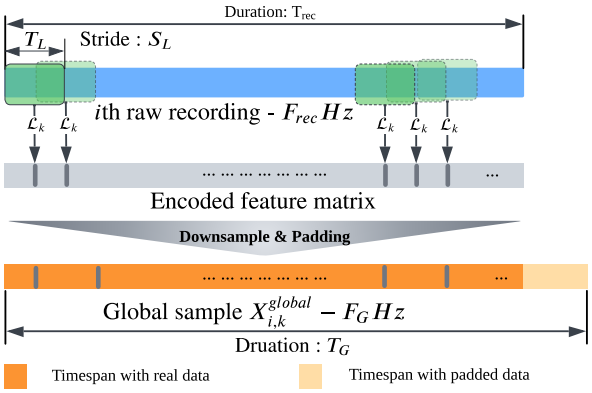


Figure 5.2: Detail of generating a global sample $X_{i,k}^{\text{global}}$ from the i-th raw recording using the k-th local model $\mathcal{L}_k$. $\mathcal{L}_k$ is trained with randomized initialization and data batching.

over the entire temporal span and identify context-dependent activity patterns.

5.1.3 SCOPE: SynthetiC glObal Pattern augmEntation

Here we introduce more insight into our proposed data augmentation technique SCOPE.

While our two-stage framework provides the foundational structure for learning global activity patterns, a major bottleneck arises from the limited number of available global training samples. This leads us to develop a new data augmentation technique, namely SCOPE, to enhance global pattern diversity in terms of reasonable sparse activity distributions.

SCOPE leverages the inherent *model volatility* we observed when training local detectors, and uses it as a stochastic source of semantic variation in global feature representations. Rather than applying random perturbations to raw sensor values, SCOPE perturbs the semantic-level representation by retraining the local model multiple times under controlled randomness.

5.1.3.1 Model Volatility

During our research, we observed fluctuations in model performance when repeatedly training models using the same data and architecture. These fluctuations originate from inherent uncertainties in problem definition for many human activity detection tasks. For instance, defining the temporal boundaries of complex activities, such as sleep stages or brief gestures, within a continuous stream of various behaviors, presents challenges prone to errors, even among human experts. Such ambiguity impedes the capacity of a model to achieve optimal alignment with the problem at

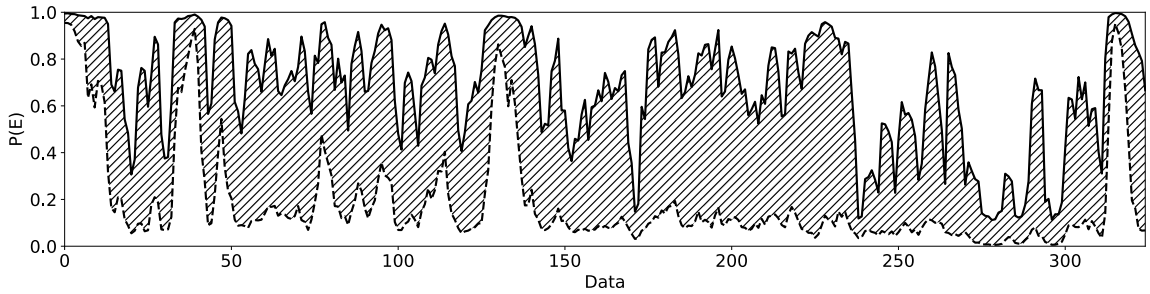


Figure 5.3: Difference (shading) between the minimum (lower dashed line) and maximum (upper solid line) estimations on probabilities of eating from 565 model instances trained from random starting points using the structure published by Sharma et al. [86]. The horizontal axis represents time, and the vertical axis represents magnitude for probabilities.

hand. Moreover, due to problem ambiguity, dataset annotators must make subjective judgments during ground truth labeling, introducing bias and further complicating the modeling process for the problem within a given dataset. Besides, repeating training models multiple times on these complex problems produces varying problem representations due to random factors such as initial states, data sequences, and stochastic optimization processes. We collectively term these variations as "model volatility."

We exploit this volatility not as a source of noise, but as a mechanism to produce semantically valid but distinct global feature sequences. By using $\mathcal{L}_1, \ldots, \mathcal{L}_M$ to encode the same raw recording X_i^{rec}, we obtain multiple distinct global representations $X_{i,k}^{\text{global}}$, each reflecting different inductive biases from training.

Figure 5.3 shows an example of model volatility and how it introduces significant variations in global patterns derived from the same raw data. These variations maintain realistic interpretations of raw sensor recordings, unlike existing augmentation methods that add random noise to individual data points independently.

5.1.3.2 Optional Strategies besides Multiple Trainings

In addition to repeating training runs with standard stochastic sources (e.g., random initialization, batch ordering), we explore two additional strategies to maximize representational diversity.

Varying sequence offsets: In sequence-to-sequence (seq2seq) models, the prediction of each timestep can be conditioned on different positions within a window. Let r for $r \in [1, R]$ denote

a valid window offset (i.e., an index for the window left), then the encoder behavior becomes:

$$X_{i,k,r}^{\text{global}} = \left[\mathcal{L}_{k,r}(X_{i,1}^{\text{local}}), \ldots, \mathcal{L}_{k,r}(X_{i,N_w}^{\text{local}})\right] \quad (i \in [1, N], k \in [1, M], r \in [1, R]) \tag{5.7}$$

which collectively expand the size $|\mathcal{D}_G|$ from N to $M \times N \times R$.

Varying training subsets: Let $\mathcal{D}_L^{(s)} \subset \mathcal{D}_L$ denote a selected subset of training windows. We train models $\mathcal{L}_m^{(s)}$ on different $\mathcal{D}_L^{(s)}$ via inner cross-validation or masking strategies. This is especially useful when the number of recordings N is small (e.g., $N = 39$ in Sleep-EDF-2013). By sampling different subsets of recordings or entirely omitting some from each training run, we force the local model to converge to alternative approximations of the activity structure.

These strategies can be combined with base SCOPE to produce a large and diverse global dataset.

SCOPE thus enhances both the *quantity* and *semantic diversity* of training samples for $\mathcal{G}$, facilitating better generalization without resorting to label-preserving signal-level noise. Experimental results in Section 5.4 demonstrate the impact of SCOPE on performance across multiple sparse-activity-detection tasks.

5.2 Experiments

Our proposed method has versatile applications in human activity recognition across various fields. In this chapter, we select four diverse test cases, as summarized in Table 5.1 and Table 5.2, to comprehensively evaluate the performance of our proposed global-pattern-analysis framework.

These test cases encompass four popular modalities: IMU, video, EEG, and audio. The recording lengths for these cases span from several seconds to several hours, encompassing a wide range of activities. This selection of test cases enables us to demonstrate the effectiveness of our method across a broad spectrum of scenarios.

These test cases share common characteristics that make them suitable for assessing our global pattern analysis framework:

- **Thematic alignment**: the recordings for each case are tailored to specific activity themes, which implies that global patterns tend to exhibit alignment between different recordings and

Table 5.1: Scenarios in Test Cases

Case	Task	Modality	Theme	Dataset	Dataset size
Meal detection	Detect and localize meal episodes from human daily activities.	IMU	All-day free-living	Clemson All-day (CAD) [86]	354 recordings, 4,680 total hours.
Intake gesture detection	Detect and localize bite and drink gestures within meal activities.	Video	Indoor meal	Clemson Cafeteria [97]	486 recordings, 107 total hours.
Sleep stage detection	Detect and localize five sleep stages from human sleep periods.	EEG	Night sleep	Sleep-EDF (Cassette) [53]	V2013: 39 recordings, 352 total hours. V2018: 153 recordings, 1,629 total hours.
Speech & music detection	Detect and localize speech & music activities from TV shows.	Audio	online TV shows	The TV Speech and Music (TVSM) [49]	Training set (TVSM-pseudo): 2,549 recordings, 1,522 total hours.

subjects within the same case.

- **Event sparsity**: events of interest either represent only a small portion of the total recording duration or are scattered sparsely, as shown in Table 5.2.
- **Recording length**: each recording is too long for traditional window-based analysis to learn long-range context due to computational constraints.
- **Limited dataset size**: current datasets in these four fields are limited in size due to the extensive labor involved in data collection and labeling.

Our framework is designed to provide a viable solution for leveraging global cues in activity recognition, and address issues related to computational burden and data scarcity.

All test cases are formulated using our framework illustrated in Figure 5.1 and Figure 5.2, with hyperparameters detailed in Table 5.3. All trainable parameters in local models are initialized from a Gaussian distribution $\mathcal{N}(0, 0.2^2)$

Let $\mathcal{L}_m$ denote the local detector trained in the m-th SCOPE iteration, where $m \in [1, M]$. From each raw recording X_i^{rec}, SCOPE generates R global samples per model instance, indexed by $X_i^{\text{global},(m,r)}$ for $r \in [1, R]$. Thus, each recording yields up to $M \times R$ global samples. All global samples are padded to a fixed length T_G and masked accordingly.

In the following subsections, we describe the four test cases (as summarized in Table 5.1 and Table 5.2) in detail, including problem descriptions, local model selection, augmentation procedure, and evaluation metrics. Global detectors are shallow LSTM-based classifiers for the purpose of validating the proposed global-pattern pipeline.

Table 5.2: Activities in Test Cases

Case	Background class	Events of interests	Total event duration	Ratio of total event duration to total recording duration
Meal detection	Other daily behaviors, such as walk, rest, etc.	Meal episodes	249 hr	1 : 18.8
Intake gesture detection	Other eating behaviors, such as utensiling, rest, etc.	Drink gestures Bite gestures	12.5 hr 5.2 hr	1 : 20.6 1 : 8.6
Sleep stage detection (Sleep-EDF V2013 dataset)	Awake	N1 N2 N3 REM	23 hr 148 hr 48 hr 64 hr	1 :15.1 1 : 2.4 1 : 7.4 1 : 5.5
Sleep stage detection (Sleep-EDF V2018 dataset)	Awake	N1 N2 N3 REM	179 hr 576 hr 109 hr 215 hr	1 : 9.1 1 : 2.8 1 : 15.0 1 : 7.6
Speech & music detection (TVSM-pseudo dataset)	Other content in TV shows	Speech Music	981 hr 953 hr	1 : 1.6 1 : 1.6

5.2.1 Case 1: Meal Detection

This case refers to detecting meal episodes from day-long IMU recordings. Automated meal detection is a focal point of healthcare research due to its critical role in measuring daily energy intake, particularly for treating obesity and related conditions like diabetes [71]. Wristbands equipped with IMU sensors have emerged as a convenient tool for obtaining all-day measurements of eating behavior with minimal user burden [22, 56, 77].

In this chapter, we utilize the Clemson All-day (CAD) dataset, the largest collection of free-living IMU recordings designed for meal detection. This dataset comprises $n = 354$ recordings sampled at $f_{\text{rec}} = 15$ Hz and includes 351 participants [86]. After data preprocessing, we identified a total of 1,133 distinct eating episodes, accounting for 249 hours of eating data out of 4,680 total hours.

We employ two SOTA neural network models in the field as local detectors $\mathcal{L}$. The first is a CNN-based model by Sharma et al., specifically designed to determine if a 6-minute window contains eating behavior or not [86]. Given the absence of published neural networks tailored for meal detection, we develop a second local model by transferring the SOTA bite detection model from Heydarian et al. [42], pre-trained on the OREBA dataset [77]. This transfer is conducted following the steps outlined in our previous work, which concentrated on enhancing meal detection in an application-oriented context (as opposed to the present chapter, which centers on developing

Table 5.3: Experimental Hyperparameters Used in Each Test Case

Parameter	Description	Meal detection	Intake gestures detection	Sleep stage detection	Speech & Music detection
F_L	Local sampling rate	15 Hz	8 Hz (CNN-LSTM), 5 Hz (X3D-S)	100 Hz	31.25 Hz
T_L	Local window length	6 min	2 sec	30 sec	20 sec
S_L	Local window stride	100 s	0.5 s	30 s	20 s
M	No. of SCOPE iterations	565	10	40 (v2013), 10 (v2018)	60
R	No. of position offsets[a]	1	16 (CNN-LSTM), 10 (X3D-S)	1	1
K'	Inner train folds[b]	1	1	4	1
N	Raw recordings	354	486	39 (v2013), 153 (v2018)	2293
N_w	Local windows per recording	850	4920	2700	200
$\lvert\mathcal{D}_G\rvert$	Total global samples	199,710	77,760, 48,600	6,240 (v2013), 6,120 (v2018)	137,580
F_G	Global sampling rate	1/100 Hz	2 Hz	1/30 Hz	5 Hz
T_G	Global sequence length	23.6 hr	41 min	22.5 hr	66.6 min
LSTM config	Layers × units	1 × 16	1 × 64	1 × 64	2 × 64
Eval. scheme	Metric type	Time + Episode	Episode	Episode	Time

[a] $R = 1$ implies seq2one output; otherwise, multiple offset-based seq2seq outputs are used.
[b] Used in inner-loop cross-validation to introduce model volatility during SCOPE.

a versatile global pattern learning framework) [99]. The resulting model works on the same window length of 6 minutes. Our training and testing configurations closely follow those detailed in previous research [86, 99].

To generate global samples X_i^{global} from the ith IMU recording, each trained local model is slid with a stride $S_L = 100$ seconds. Each global sample X_i^{global} is zero-padded to length $T_G = 850$. Padding is masked during loss calculation and evaluation. SCOPE is applied by training each local model $M = 565$ times from different random initializations, producing $M = 565$ global samples per recording. Thus, $|\mathcal{D}_G| = 565 \times 354 = 199,710$ global samples.

Our evaluation process employs 5-fold subject-wise cross-validation as in [86]. We report various metrics, including time-based (TPR, TNR, F1, precision, and weighted accuracy) and episode-based (TPR and FP/TP) measures, in accordance with the evaluation scheme established in previous works [86, 99].

5.2.2 Case 2: Intake Gesture Detection

This case refers to recognizing bite and drink gestures within meal video recordings. The automatic detection of in-meal intake behaviors plays a pivotal role in quantifying eating habits objectively, offering valuable insights for targeted dietary interventions [78]. Unlike sensors that often rely on users wearing devices, cameras provide a non-contact method for collecting video data during meals, making them particularly useful for this task.

For our experiments, we employ the Clemson Cafeteria dataset, which represents the largest collection of in-meal videos designed for gesture detection. This dataset comprises 486 videos from 264 participants, encompassing 12.5 hours of bite gestures, 5.2 hours of drink gestures, and 89.4 hours of other behaviors [97, 98].

We utilize two SOTA models in the field as local detectors $\mathcal{L}$: CNN-LSTM with ResNet-34 as the backbone, derived from the work by Heydarian et al. [75], and X3D-S, based on the work of Feichtenhofer [26]. Both models operate on a 2-second window.

We modify these models to maintain the temporal axis within and produce frame-level predictions in a sequence-to-sequence style, following the approach detailed in our previous work [98] which primarily focused on enhancing intake gesture detection for dietary monitoring systems (in contrast to the current chapter, which emphasizes the creation of a versatile global pattern learning framework). This modification allows us to encode a recording multiple times in a single pass by shifting the prediction position within each window. This trick significantly reduces the number of training runs during SCOPE, and saves considerable training time for complex video models. Depending on the model input frequency, the number of prediction position offsets per model is $R = 16$ for CNN-LSTM and $R = 10$ for X3D-S.

To generate global samples X_i^{global} from the ith video recording, each trained local model is moved with a stride $S_L = 0.5$ seconds. Each global sample is padded to $T_G = 5000$ (about 41 minutes). Padding is masked during training and evaluation. SCOPE is applied by training each local model $M = 10$ times using random initializations, yielding $M \times R$, which is 160 for CNN-LSTM and 100 for X3D-S, global samples per recording.

During evaluation, we randomly partition the participants in the Clemson Cafeteria dataset into training and test sets with an 80% to 20% ratio. We report episode-based metrics (TPR and FP/TP) for performance comparison before and after applying a global detector, following the same

evaluation scheme established in prior studies [58, 98].

5.2.3 Case 3: Sleep Stage Detection

In this case, we focus on detecting sleep stages from whole-night electroencephalogram (EEG) recordings. Disruptions in sleep patterns and structure can significantly impact daily routines and public safety [25]. The American Academy of Sleep Medicine (AASM) defines six sleep stages: awake (W), non-rapid eye movement (N-REM) stage 1 (N1), N-REM stage 2 (N2), N-REM stage 3 (N3), and rapid eye movement (REM) [7]. Automated sleep stage scoring systems play a vital role in interpreting physiological signals to assess sleep stage information.

In this chapter, we employ the widely used Sleep-EDF-v2013 dataset and its expanded version, Sleep-EDF-v2018 [53], to evaluate our pipeline in the sleep-stage-detection domain. We use recordings from the Sleep Cassette Study, which did not involve sleep-related medication, to reduce irregular patterns and provide a fair evaluation of our pipeline. V2013 includes $n = 39$ recordings totaling 352 hours, and V2018 includes $n = 153$ recordings totaling 1,629 hours.

Following the same dataset usage as in [24, 82, 95], we use the Fpz-cz channel in the EEG signal for analysis. We exclude any UNKNOWN stages and W stages that are more than 30 minutes before or after sleep periods, and merge stages N3 and N4 into one stage (N3) to align with the AASM standard.

We utilize one of SOTA models in the field, AttnSleep [24], as our local model $\mathcal{L}$. After re-implementation, we found that many recently published sleep stage models did not provide SOTA performance as claimed, and AttnSleep demonstrated the best performance up to our evaluation date. The model operates on a 30-second window of EEG signals and outputs a probability value for each class.

To create global samples X_i^{global} from the ith night EEG recording , each local model is slid with stride $S_L = 30$ seconds over each recording. Each global sample is padded to $T_G = 2700$ timesteps (approx. 22.5 hours). Padding is masked during training and evaluation.

Due to the limited number of recordings in both Sleep-EDF dataset versions, the data diversity of global samples is severely constrained. To address this issue, we apply nested cross-validation with outer and inner folds of 5 and 4, respectively, to increase the diversity of training data and encourage model volatility during SCOPE. Specifically, we divide the dataset into 5 outer folds, with 4 folds for training and 1 fold for testing. Within each training set, we create 4 inner-folds,

using 3 for training local models.

Thus, after training each local model $M = 10$ times on v2018 (or $M = 40$ times on v2013), each training split from the outer cross-validation results in $M \times 4 = 40(160)$ trained model instances. By introducing this volatility in the training data, we expect model weights in local models to differ more between different training runs.

During evaluation, we report episode-based metrics, including F1-score, precision, recall, and macro-averaged F1-score (MF1), for performance comparison before and after applying the stage 2 global detector, following the same evaluation scheme as in [24].

5.2.4 Case 4: Speech and Music Detection

This task involves segmenting speech and music events from the timespan of a broadcast audio recording. This segmentation aids in the creation of metadata related to the content and often serves as a preprocessing step for semantic-level tasks such as speech and lyrics transcription and content recognition [106].

In this chapter, we utilize the largest dataset with long audio recordings, namely the TVSM dataset [49], to evaluate our pipeline for speech and music detection. The dataset comprises professionally recorded and produced audio from TV shows. We exclude audio recordings less than 20 minutes in length from training set to eliminate short broadcast fragments without sufficient global context. This results in $n = 2293$ audio recordings in the training set (TVSM-pseudo as in [49]) and 20 audio recordings in the test set (TVSM-test as in [49]). We use log-Mel spectrogram and per-channel energy normalization (PCEN) as raw audio features, as this combination yielded the best performance in [49].

We employ two SOTA models in the field, namely TCN and CRNN, as our local models $\mathcal{L}$, as shown in [49]. Both models operate on a 20-second window and output per-class probabilities for every frame in the input (approximately 31 frames per second).

To construct global samples X_i^{global} from the ith recording, we slide each local model with stride $S_L = 20$ seconds. Outputs are downsampled to 5 frames per second to reduce sequence length. Each global sample is zero-padded to $T_G = 20{,}833$ frames (equivalent to 4000 seconds). The padded part is masked and excluded from loss calculation and model evaluation. We observe that only a few global samples in the training set are longer than 4,000 seconds, and we clip them at the tails to match the length.

During evaluation, we report time-based metrics (segment-level) at a resolution of 5 frames per second, including error rate, deletion rate, insertion rate, F1-score, precision, and recall for performance comparison before and after applying the stage 2 global detector, following the same evaluation scheme as in [49].

5.2.5 Global Detector

For validation purposes, we leverage a simple yet effective RNN-based model as the stage 2 global detector for each test case. Its objective is to demonstrate the utility of incorporating global sequence context from full recordings constructed via our framework.

The architecture consists of one or two bidirectional Long Short-Term Memory (BiLSTM) layers for temporal modeling, followed by a position-wise fully connected (dense) layer to make categorical predictions for each position in a global sample. Specifically, the model performs the following transformation:

$$\mathcal{G}(X_i^{\text{global}}) = \text{Softmax}(W_o H_i + b_o) \tag{5.8}$$

where $H_i = \text{BiLSTM}(X_i^{\text{global}})$ is the hidden representation sequence, and W_o, b_o are weights and biases of the dense layer applied to each timestep.

We vary the configurations of LSTM layers to optimize performance for each test case. For Case 1, we use one LSTM layer with 16 units. For Case 2 and Case 3, we utilize one LSTM layer with 64 units. For Case 4, we employ a stack of two LSTM layers with 64 units.

Each global detector is trained for 100 epochs using the Adam optimizer [54]. The learning rate is set to 0.001 with an exponentially decaying rate of 0.98.

We follow the same training/test splits used for the corresponding local models in each case. For testing, all available model instances $\mathcal{L}_k$, $k \in [1, m]$ are used to generate global samples from the test set. We evaluate the performance of a global detector on all generated global samples and report average performance metrics.

5.3 Results

In this section, we present the results of our experiments, aiming to evaluate the performance of our proposed global pattern analysis framework across four different test cases For each test case,

Table 5.4: Case 1: Performance for meal detection (IMU dataset: CAD). Bold numbers indicate better results.

Method	Time					Episodes	
	TPR	TNR	F1	Precision	WAcc	TPR	FP/TP
Heydarian et al. [42]	0.54	0.65	0.14	0.08	0.60	0.59	3.0
Our framework + Heydarian et al. [42]	**0.60**	**0.82**	**0.25**	**0.15**	**0.71**	**0.71**	3.0
Sharma et al. [86]	0.76	0.86	0.35	0.23	0.81	0.87	1.9
Our framework + Sharma et al. [86]	**0.81**	**0.86**	**0.38**	**0.25**	**0.84**	**0.89**	**1.4**

Table 5.5: Case 2: Performance for intake gesture detection (video dataset: Clemson Cafeteria). Bold numbers indicate better results.

	Episode F1		Episode Precision		Episode Recall	
	Bite	Drink	Bite	Drink	Bite	Drink
CNN-LSTM [75]	0.89	0.67	0.86	0.58	0.92	0.79
Our framework + CNN-LSTM [75]	**0.94**	**0.86**	**0.97**	**0.90**	0.92	**0.83**
X3D-S [26]	0.91	0.73	0.88	0.68	**0.93**	0.80
Our framework + X3D-S [26]	**0.93**	**0.88**	**0.95**	**0.95**	0.91	**0.81**

we provide a comparative analysis of local models and global detectors.

5.3.1 Case 1: Meal Detection

Table 1 presents the performance evaluation results for meal detection using IMU data from the CAD dataset. Our framework, when combined with either window-based model, exhibited substantial performance improvements across most time-based metrics. At the episode level, our framework improved Sharma et al. model by a margin of 0.5 in episode FP/TP while maintaining a similar episode TPR. Additionally, when wrapping the transferred model, our framework achieved a 0.12 increase in episode TPR while maintaining a comparable episode FP/TP ratio. These results underscore the effectiveness of our global context analysis approach in achieving precise meal detection, which holds significant promise for applications in dietary monitoring and healthcare.

5.3.2 Case 2: Intake Gesture Detection

Table 4 showcases the outcomes of our intake gesture detection experiments conducted on Clemson Cafeteria dataset. For CNN-LSTM, our framework achieved an improved F1 score with a margin of 0.05 for bite gestures, accompanied with an impressive margin of 0.19 for drink gestures, resulting in macro-averaged F1 (MF1) of 0.83. Similar improvement can be seen on X3D-S. Drink gestures distribute much coarser than bite gestures, leading to increased detection difficulty for local models as most windows do not contain drink gestures. The difficulty is diminished by looking global

context and analyzing gesture distributions and inter-gesture correlations. Besides, we can see that our global-level framework improved window-based models mostly on episode precisions, indicating the strong ability of reducing false triggers which could be fake gestures or transient false detections due to limited information within windows.

5.3.3 Case 3: Sleeping Stage Detection

Table 5.6: Case 3: Performance for sleeping stage detection (EEG dataset: Sleep-EDF). Bold numbers indicate better results.

Dataset	Method	Per-class Episode F1					Overall
		W	N1	N2	N3	REM	MF1
Sleep-EDF-2013	AttnSleep [24]	0.85	0.39	0.87	0.88	0.75	0.75
	Our framework + AttnSleep [24]	**0.88**	**0.50**	**0.88**	**0.88**	**0.85**	**0.80**
Sleep-EDF-2018	AttnSleep [24]	0.91	0.41	0.82	**0.80**	0.69	0.73
	Our framework + AttnSleep [24]	**0.92**	**0.52**	**0.85**	0.79	**0.82**	**0.78**

In our sleep stage detection experiments conducted on the Sleep-EDF dataset, as depicted in Table 5, we employed the AttnSleep model as the local classifier and observed notable improvements in performance when integrated with our global context analysis framework. Our framework achieved 0.5 better on overall macro-averaged F1 (MF1) score on both datasets, demonstrating the effectiveness of our approach in sleep stage classification. Performance improvements mostly lie on W, N1 and REM classes. Among the rest classes, N2 class is the major class in datasets [24, 95] which reveals the least coarse context in long-term span, and N3 class is the mixture of N3 and N4 classes originally labeled according to R&K rules.

5.3.4 Case 4: Speech and Music Detection

Table 5.7: Case 4: Performance for segment-level speech and music detection (audio dataset:TVSM). Bold numbers indicate better results.

Method	Error rate		Deletion Rate		Insertion rate		F1		Precision		Recall	
	Music	Speech	Music	Speech	Music	Speech	Music	Speech	Music	Speech	Music	Speech
TCN [49]	0.18	0.19	0.07	0.05	0.10	0.14	0.913	0.910	0.900	0.871	0.926	0.953
Our framework + TCN [49]	**0.16**	**0.18**	0.07	**0.04**	**0.09**	0.14	**0.923**	**0.913**	**0.913**	**0.873**	**0.934**	**0.955**
CRNN [49]	0.19	0.18	0.09	0.04	0.09	0.14	0.905	0.913	0.905	0.870	0.905	**0.960**
Our framework + CRNN [49]	**0.15**	0.18	**0.06**	0.04	0.09	0.14	**0.926**	**0.914**	**0.914**	**0.875**	**0.938**	0.956

Table 6 shows our experimental results on segment-level speech and music detection using TVSM dataset. Our framework significantly improves the F1, precision, and recall for music events,

while achieving marginal improvements for speech events. This observation aligns with the inherent challenges of modeling global patterns. Speech events tend to occur randomly and frequently within TV shows, giving rise to elusive global patterns that transcend individual recording boundaries. Consequently, our global detector may not effectively capture extensive global-level context information beyond the local features encoded by window-based models. Conversely, music events are distributed more sparsely over time, and often follow the rhythms of TV shows or become entangled with other events, including speech before and after.

5.3.5 Result Summary

The experimental results across our four diverse test cases offer valuable insights into the effectiveness and versatility of our proposed global context analysis framework.

Firstly, our framework demonstrates the capability to extract meaningful global patterns from data with varying time lengths, event durations, types, and modalities, even when training data is limited. Across all cases, it consistently enhances SOTA models across a range of performance metrics.

Secondly, Our approach improves performance on both time-based metrics, as seen in cases 1 and 4, and episode-based metrics, as observed in cases 1 to 3. This adaptability supports the utility in diverse evaluation scenarios, accommodating different analytical requirements.

Furthermore, our framework excels in enhancing the detection of sparsely distributed classes in multi-class cases. By employing a global detector that scrutinizes the entire recording simultaneously, our framework taps into insights regarding global distributions and inter-event correlations, leading to marked improvements in identifying these less frequent classes. For example, in case 2, our global detector achieves higher precision in detecting the minority class, drink gestures, while in case 4, it demonstrates notable enhancements in precision and recall when identifying the minority class, music segments.

Lastly, the shared characteristic of improved performance, while preserving computational efficiency, is noteworthy. In this chapter, we leverage simple RNN-based models as global detectors and still make effective use of global context to improve existing SOTA local models. After training via SCOPE augmentation, the framework maintains manageable computational requirements despite the integration of global context analysis, making it practical for wrapping complex window-based models or deploying in resource-constrained applications.

5.4 Ablation Study

This section evaluates the effectiveness of the SCOPE augmentation method across four use cases, comparing it with no augmentation and a traditional augmentation method that involves adding random noise. For each setting, we train and test models 10 times and report averages and standard deviations for each metric.

For the traditional augmentation method, we add random noise to each normalized global sample X_G, where $X_G \in \mathbb{R}^{T \times C}$ represents a global sample of length T and C channels. The augmentation process is defined as

$$X'_G = X_G + \epsilon$$

where $\epsilon \in \mathbb{R}^{T \times C}$ is a noise matrix with each element $\epsilon_{t,c} \sim \mathcal{N}(0, 0.1^2)$ independently sampled for time step t and channel c. Prior to augmentation, X_G is normalized such that its elements approximately follow $\mathcal{N}(0, 1.0^2)$.

Tables 5.8, 5.9, 5.10, and 5.11 present results from meal detection, intake gesture detection, sleep stage detection, and speech and music detection, respectively.

The results from our ablation studies demonstrate that the SCOPE augmentation method offers more advantages for training global detectors over the other augmentation types.

We can see that the relative gain from SCOPE diminishes as the dataset size N increases. For instance, improvements are marginal in the speech/music detection case ($n = 2293$), while much more pronounced in smaller datasets like meal ($n = 354$) and gesture detection ($n = 486$). This observation suggests that SCOPE is most effective when data scarcity and overfitting are more prominent concerns.

Another noteworthy observation is that SCOPE reduces variance across training runs, as evidenced by lower standard deviation values. In the eating gesture detection case, the most challenging of the four prior to applying SCOPE, high standard deviations and low averages in performance metrics suggest that some models do not converge over all 10 training rounds. This instability stems from significant variability in dietary patterns and the small amount of available global samples. In such scenarios, SCOPE enhances diversity at the global pattern level, in contrast to traditional methods that introduce random perturbations at individual data points.

Table 5.8: Meal detection with different augmentation on global samples. Bold numbers indicate the best results.

Augmentation	Episode Metrics		Time Metrics				
	Episode FP/TP	Episode TPR	TPR	TNR	F1	Precision	Weighted Acc
None	1.693 ± 1.235	0.407 ± 0.243	0.388 ± 0.229	0.780 ± 0.050	0.131 ± 0.058	0.084 ± 0.039	0.584 ± 0.059
Gaussian noise	1.701 ± 0.723	0.484 ± 0.186	0.465 ± 0.174	0.697 ± 0.106	0.136 ± 0.036	0.081 ± 0.022	0.581 ± 0.060
SCOPE	**1.362 ± 0.023**	**0.871 ± 0.003**	**0.794 ± 0.005**	**0.873 ± 0.004**	**0.392 ± 0.006**	**0.261 ± 0.005**	**0.833 ± 0.002**

Table 5.9: Eating gesture detection with different augmentation on global samples. Bold numbers indicate the best results.

Augmentation	**Episode F1**		**Episode Precision**		**Episode Recall**	
	Bite	Drink	Bite	Drink	Bite	Drink
None	0.081 ± 0.138	0.068 ± 0.195	0.520 ± 0.433	0.161 ± 0.323	0.050 ± 0.091	0.054 ± 0.158
Gaussian noise	0.098 ± 0.258	0.177 ± 0.260	0.675 ± 0.444	0.411 ± 0.419	0.084 ± 0.235	0.146 ± 0.227
SCOPE	**0.938 ± 0.001**	**0.851 ± 0.002**	**0.952 ± 0.002**	**0.914 ± 0.005**	**0.924 ± 0.003**	**0.797 ± 0.003**

5.5 Conclusion

In conclusion, our novel framework for learning global patterns offers a practical and versatile solution to enhance the recognition of sparse human activities within lengthy sensor data recordings. We demonstrate the effectiveness of our framework across four various application domains, including different sensor data types, and recording lengths. Additionally, our innovative augmentation method (SCOPE) enhances the ability of the framework to learn global patterns even when only a limited number of well-labeled recordings are available.

However, it is important to acknowledge several potential limitations in our approach, which offer promising avenues for future research. One such limitation arises from the separation between the training of local feature detectors and the global pattern detector Future work may explore tighter integration of these two networks, enabling the model to learn global patterns directly from raw data. Besides, our approach is intended for offline rather than real-time analysis. Future work could explore

Table 5.10: Sleep stage detection with different augmentation on global samples. Bold numbers indicate the best results.

Augmentation	Per-class Episode F1					Overall
	W	N1	N2	N3	R	
None	0.907 ± 0.004	0.427 ± 0.018	0.808 ± 0.010	0.448 ± 0.172	0.732 ± 0.037	0.664 ± 0.020
Gaussian noise	0.909 ± 0.004	0.408 ± 0.027	0.809 ± 0.008	0.482 ± 0.144	0.737 ± 0.034	0.669 ± 0.015
SCOPE	**0.924 ± 0.001**	**0.518 ± 0.003**	**0.847 ± 0.001**	**0.794 ± 0.002**	**0.819 ± 0.002**	**0.781 ± 0.010**

Table 5.11: Speech and music detection with different augmentation on global samples. Bold numbers indicate the best results.

Augmentation	F-measure		Precision		Recall	
	Speech	Music	Speech	Music	Speech	Music
None	0.912 ± 0.001	0.926 ± 0.001	0.870 ± 0.002	0.916 ± 0.002	**0.958 ± 0.001**	0.937 ± 0.003
Gaussian noise	0.890 ± 0.014	0.918 ± 0.006	0.840 ± 0.012	**0.921** ± 0.007	0.947 ± 0.022	0.914 ± 0.014
SCOPE	**0.913 ± 0.000**	**0.928 ± 0.001**	**0.873 ± 0.001**	0.916 ± **0.002**	0.957 ± 0.001	**0.940 ± 0.003**

embedding the information learned via global pattern analysis into a real-time framework. Another possible research direction concerns the proposed SCOPE augmentation method, which generates new patterns based on intermediate features. Future research may explore the creation of synthetic raw recordings from these augmented feature-level global patterns, possibly incorporating methods like diffusion models [4, 46].

Chapter 6

Conclusion and Future Directions

This dissertation develops a unified framework for sparse-event recognition in long recordings. The central conclusion is that detection becomes more reliable when local evidence is modeled jointly with full-recording temporal context, rather than by isolated sliding-window predictions alone.

At the methodological level, the dissertation addresses a common problem setting: events of interest are brief or infrequent, recordings are long, and labeled datasets are limited. In this setting, local-only models repeatedly encounter two failure modes: class-imbalance bias and context-free false alarms near ambiguous transitions. The proposed framework addresses both by explicitly modeling temporal structure at two scales and by reframing sparse-event detection as a recording-level global-context problem.

The pipeline is organized in two stages. Stage one uses efficient window-based encoders to extract local probabilities or features from short segments. Stage two treats the full sequence of stage-one outputs as a recording-level sample and learns global temporal patterns over the complete timespan. This decomposition extends the effective temporal field of view from local windows to full recordings while keeping computational requirements practical for high-dimensional modalities such as video.

To address limited sample size at the recording level, the framework introduces SCOPE (SynthetiC glObal Pattern augmEntation) between stages. By leveraging stage-one model variability and offset-induced uncertainty, SCOPE generates multiple feature-level realizations of each recording without additional manual annotation. Across our studies, this mechanism increases stage-two sample diversity, improves optimization stability, and reduces sensitivity to overfitting in small

datasets.

Empirically, the dissertation validates this framework through completed dietary-monitoring and cross-domain studies. In dietary monitoring, day-length wrist-motion analysis improves eating-episode detection [99], and meal-length video analysis improves intake-gesture recognition through global temporal modeling [98]. A dedicated dataset chapter further contributes a realistic cafeteria video benchmark that supports robust evaluation and model development for sparse intake gestures. Together, these chapters demonstrate transferability across sensing modalities and event granularities while preserving the practical advantage of wrapping existing local detectors. They also show that top-down global analysis complements, rather than replaces, fine-grained local event analysis.

This unified formulation also supports extensions beyond dietary monitoring. The dissertation includes sleep-stage detection from whole-night EEG using Sleep-EDF-v2013 and Sleep-EDF-v2018 [7, 53], and speech/music segmentation in long-form broadcast audio using TVSM [49]. These domains preserve the same sparse-event, long-context, and limited-label characteristics, further validating the same methodological story: combining reliable local evidence with global temporal reasoning to improve sparse-event recognition in realistic long recordings.

The broader framework study consolidates these conclusions across four test cases: meal detection, intake-gesture detection, sleep-stage detection, and speech/music segmentation. Several consistent patterns emerge. First, global modeling improves both time-based and episode-based metrics when meaningful recording-level structure exists. Second, gains are often strongest for sparse minority classes (e.g., drink gestures and music), where global distributional cues and inter-event relationships are most informative. Third, improvements remain attainable with lightweight global models (e.g., RNN-based stage-two detectors), so most computational burden can remain in stage one while still benefiting from global reasoning. This supports the general framework claim that local evidence and global context should be modeled jointly but with different computational roles.

Ablation results clarify when augmentation is most beneficial. Compared with no augmentation or simple noise perturbation, our new data augmentation method consistently improves global-detector training and typically reduces run-to-run variance, with the largest gains in smaller datasets where stage-two overfitting is most severe. This supports a key thesis claim: for long-recording tasks, recording-level data efficiency is as important as model architecture.

Several limitations motivate future work. First, local and global components are currently

trained separately, so stage-two performance remains bounded by stage-one representations. A natural next step is tighter end-to-end optimization that preserves the efficiency of the two-stage design while enabling joint representation learning. Second, the current framework is primarily offline; future work should explore how offline global-context priors can be distilled into low-latency, streaming-capable models for real-time deployment. Third, SCOPE currently augments in feature space; future work may investigate principled generation of synthetic raw recordings or raw-label trajectories, including diffusion-based sequence generation, to improve robustness under extreme data scarcity [4, 46].

Bibliography

[1] Geoffrey Appelboom, Esteban Camacho, and Michael Abraham. Smart wearable body sensors for patient self-assessment and monitoring. *Archives of Public Health*, 72(1):1–9, 2014.

[2] M Carolina Archundia Herrera and Catherine B Chan. Narrative review of new methods for assessing food and energy intake. *Nutrients*, 10(8):1064, 2018.

[3] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[4] German Barquero, Sergio Escalera, and Cristina Palmero. Belfusion: Latent diffusion for behavior-driven human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2317–2327, 2023.

[5] France Bellisle. Impact of the daily meal pattern on energy balance. *Scandinavian Journal of Nutrition*, 48(3):114–118, 2004.

[6] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9592–9600, 2019.

[7] Richard B Berry, Rita Brooks, Charlene E Gamaldo, Susan M Harding, Carole L Marcus, and Bradley V Vaughn. The aasm manual for the scoring of sleep and associated events. *Rules, Terminology and Technical Specifications, Darien, Illinois, American Academy of Sleep Medicine*, 176:2012, 2012.

[8] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. Wiley, 5th edition, 2015.

[9] Andrew W Brown, Stella Aslibekyan, Dennis Bier, Rafael Ferreira da Silva, Adam Hoover, David M Klurfeld, Eric Loken, Evan Mayo-Wilson, Nir Menachemi, Greg Pavela, Patrick D Quinn, Dale Schoeller, Carmen Tekwe, Danny Valdez, Colby J Vorland, Leah D Whigham, and David B Allison. Toward more rigorous and informative nutritional epidemiology: The rational space between dismissal and defense of the status quo. *Critical Reviews in Food Science and Nutrition*, pages 1–18, 2021.

[10] Lora E Burke, Jing Wang, and Mary A Sevick. Self-monitoring in weight loss: a systematic review of the literature. *Journal of the American Dietetic Association*, 111(1):92–102, 2011.

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[12] Catherine M Champagne, George A Bray, and Anny H Kurtz. Energy intake and energy expenditure: a controlled study comparing dietitians and non-dietitians. *Journal of the American Dietetic Association*, 102(10):1428–1432, 2002.

[13] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[14] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)*, 54(4):1–40, 2021.

[15] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.

[16] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[17] Tommy Chou, Adam W Hoover, Stephanie P Goldstein, Dante Greco-Henderson, Corby K Martin, Hollie A Raynor, Eric R Muth, and J Graham Thomas. An explanation for the accuracy of sensor-based measures of energy intake: Amount of food consumed matters more than dietary composition. *Appetite*, 194:107176, 2024.

[18] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, 108:107561, 2020.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[20] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634, 2015.

[21] Yujie Dong, Adam Hoover, Jenna Scisco, and Eric Muth. A new method for measuring meal intake in humans via automated wrist motion tracking. *Applied Psychophysiology and Biofeedback*, 37(3):205–215, 2012.

[22] Yujie Dong, Jenna Scisco, Mike Wilson, Eric Muth, and Adam Hoover. Detecting periods of eating during free-living by tracking wrist motion. *IEEE journal of biomedical and health informatics*, 18(4):1253–1260, 2013.

[23] Abul Doulah, Tonmoy Ghosh, Delwar Hossain, Masudul H Imtiaz, and Edward Sazonov. Automatic ingestion monitor version 2 - a novel wearable device for automatic food intake detection and passive capture of food images. *IEEE Journal of Biomedical and Health Informatics*, 25(2):568–576, February 2021.

[24] Emadeldeen Eldele, Zhenghua Chen, Chengyu Liu, Min Wu, Chee-Keong Kwoh, Xiaoli Li, and Cuntai Guan. An attention-based deep learning approach for sleep stage classification with single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:809–818, 2021.

[25] Oliver Faust, Hajar Razaghi, Ragab Barika, Edward J Ciaccio, and U Rajendra Acharya. A review of automated sleep stage scoring based on physiological signals for the new millennia. *Computer methods and programs in biomedicine*, 176:81–91, 2019.

[26] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020.

[27] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[28] Jason G Fleischer, Sai Krupa Das, Manjushri Bhapkar, Emily N C Manoogian, and Satchidananda Panda. Associations between the timing of eating and weight-loss in calorically restricted healthy adults: findings from the calerie study. *Experimental Gerontology*, 165:111837, August 2022.

[29] Yang Gao, Ning Zhang, Honghao Wang, Xiang Ding, Xu Ye, Guanling Chen, and Yu Cao. ihear food: eating detection using commodity bluetooth headsets. In *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 163–172. IEEE, 2016.

[30] Luke Gemming, Aiden Doherty, Jennifer Utter, Emma Shields, and Cliona Ni Mhurchu. The use of a wearable camera to capture and categorise the environmental and social context of self-identified eating episodes. *Appetite*, 92:118–125, 2015.

[31] Aurelien Geron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2019.

[32] Satchin Gill and Satchidananda Panda. A smartphone app reveals erratic diurnal eating patterns in humans that can be modulated for health benefits. *Cell Express*, 22:789–798, 2015.

[33] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, and Xingyu Liu. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

[34] Nicole Gruber and Alfred Jockisch. Are gru cells more specific and lstm cells more sensitive in motive classification of text? *Frontiers in Artificial Intelligence*, 3:40, 2020.

[35] Neha Gupta, Suneet K Gupta, Rajesh K Pathak, Vanita Jain, Parisa Rashidi, and Jasjit S Suri. Human activity recognition in artificial intelligence framework: A narrative review. *Artificial intelligence review*, 55(6):4755–4808, 2022.

[36] JL Guss and HR Kissileff. Microstructural analyses of human ingestive patterns: from description to mechanistic hypotheses. *Neuroscience & Biobehavioral Reviews*, 24(2):261–268, 2000.

[37] Craig M Hales. *Prevalence of Obesity and Severe Obesity Among Adults:*. US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics, 2020.

[38] Yue Han, Sri Kalyan Yarlagadda, Tonmoy Ghosh, Fengqing Zhu, Edward Sazonov, and Edward J Delp. Improving food detection for images from a wearable egocentric camera. *Electronic Imaging*, 2021(8):286–1, 2021.

[39] John A Hawley, Paolo Sassone-Corsi, and Juleen R Zierath. Chrono-nutrition for the prevention and treatment of obesity and type 2 diabetes: From mice to men. *Diabetologia*, 63(11):2253–2259, 2020.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[41] Hamid Heydarian, Marc Adam, Tracy Burrows, Clare Collins, and Megan E Rollo. Assessing eating behaviour using upper limb mounted motion sensors: A systematic review. *Nutrients*, 11(5):1168, 2019.

[42] Hamid Heydarian, Philipp V Rouast, Marc TP Adam, Tracy Burrows, Clare E Collins, and Megan E Rollo. Deep learning for intake gesture detection from wrist-worn inertial sensors: The effects of data preprocessing, sensor modalities, and sensor positions. *IEEE Access*, 8:164936–164949, 2020.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[44] Steve Hodges, Lyndsay Williams, Emma Berry, Shahram Izadi, James Srinivasan, Alex Butler, Gavin Smyth, Narinder Kapur, and Ken Wood. Sensecam: A retrospective memory aid. In *International Conference on Ubiquitous Computing*, pages 177–193. Springer, 2006.

[45] Adam Hoover. Data description: Clemson cafeteria dataset. `http://cecas.clemson.edu/~ahoover/cafeteria/`, 2020.

[46] Tobias Höppe, Arash Mehrjou, Stefan Bauer, Didrik Nielsen, and Andrea Dittadi. Diffusion models for video prediction and infilling. *arXiv preprint arXiv:2206.07696*, 2022.

[47] Delwar Hossain, Tonmoy Ghosh, and Edward Sazonov. Automatic count of bites and chews from videos of eating episodes. *IEEE Access*, 8:101934–101945, 2020.

[48] Qianyi Huang, Wei Wang, and Qian Zhang. Your glasses know your diet: Dietary monitoring using electromyography sensors. *IEEE Internet of Things Journal*, 4(3):705–712, 2017.

[49] Yun-Ning Hung, Chih-Wei Wu, Iroro Orife, Aaron Hipple, William Wolcott, and Alexander Lerch. A large tv dataset for speech and music activity detection. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1):21, 2022.

[50] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[51] Hamid Kalantarian, Nabil Alshurafa, and Majid Sarrafzadeh. A survey of diet monitoring technology. *IEEE Pervasive Computing*, 16(1):57–65, 2017.

[52] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, and Paul Natsev. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[53] Bob Kemp, Aeilko H. Zwinderman, Bert Tuk, Hilbert A. C. Kamphuisen, and Josefien J. L. Oberye. Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg. *IEEE Transactions on Biomedical Engineering*, 47(9):1185–1194, 2000.

[54] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[55] Konstantinos Kyritsis, Christos Diou, and Anastasios Delopoulos. Modeling wrist micromovements to measure in-meal eating behavior from inertial sensor data. *IEEE Journal of Biomedical and Health Informatics*, 23(6):2325–2334, 2019.

[56] Konstantinos Kyritsis, Christos Diou, and Anastasios Delopoulos. A data driven end-to-end approach for in-the-wild monitoring of eating behavior using smartwatches. *IEEE Journal of Biomedical and Health Informatics*, 25(1):22–34, 2020.

[57] Shiyang Li, Xiyuan Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu Wang, and Xinyu Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5243–5253, 2019.

[58] Yadnyesh Y Luktuke and Adam Hoover. Segmentation and recognition of eating gestures from wrist motion using deep learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1368–1373. IEEE, 2020.

[59] Amira Ben Mabrouk and Ezzeddine Zagrouba. Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications*, 91:480–491, 2018.

[60] Neelu Madan, Nicolae-Cătălin Ristea, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised masked convolutional transformer block for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):525–542, 2023.

[61] Christopher A Merck, Christina Maher, Mark Mirtchouk, Min Zheng, Yuxiao Huang, and Samantha Kleinberg. Multimodality sensing for eating recognition. In *PervasiveHealth*, pages 130–137, 2016.

[62] Mark Mirtchouk and Samantha Kleinberg. Detecting granular eating behaviors from body-worn audio and motion sensors. In *2021 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pages 1–4. IEEE, 2021.

[63] Mark Mirtchouk, Drew Lustig, Alexandra Smith, Ivan Ching, Min Zheng, and Samantha Kleinberg. Recognizing eating from body-worn sensors: Combining free-living and laboratory data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–20, 2017.

[64] Holly L Nicastro, Susan Vorkoper, Rene Sterling, Ariella R Korn, Alison GM Brown, Padma Maruvada, and April Y Oh. Opportunities to advance implementation science and nutrition research: a commentary on the strategic plan for nih nutrition research. *Translational Behavioral Medicine*, 13(1):1–6, 2023.

[65] World Health Organization. Obesity: preventing and managing the global epidemic. *World Health Organization technical report series*, 894:1–253, 2000.

[66] World Health Organization. Noncommunicable diseases progress monitor 2020. 2020. *Geneva: WHO*, 2021.

[67] World Health Organization. Obesity and overweight. `https://www.who.int/en/news-room/fact-sheets/detail/obesity-and-overweight`, 2022. Accessed May 2026.

[68] Sebastian Päßler and Wolf-Joachim Fischer. Food intake monitoring: Automated chew event detection in chewing sounds. *IEEE Journal of Biomedical and Health Informatics*, 18(1):278–289, 2013.

[69] Sebastian Paßler, Wolf-Joachim Fischer, and Ivan Kraljevski. Adaptation of models for food intake sound recognition using maximum a posteriori estimation algorithm. In *2012 Ninth International Conference on Wearable and Implantable Body Sensor Networks*, pages 148–153. IEEE, 2012.

[70] Jianing Qiu, Frank P-W Lo, Shuo Jiang, Ya-Yen Tsai, Yingnan Sun, and Benny Lo. Counting bites and recognizing consumed food from videos for passive dietary monitoring. *IEEE Journal of Biomedical and Health Informatics*, 25(5):1471–1482, 2020.

[71] Margaret Raber, Yue Liao, Anne Rara, Susan M Schembre, Kate J Krause, Larkin Strong, Carrie Daniel-MacDougall, and Karen Basen-Engquist. A systematic review of the use of dietary self-monitoring in behavioural weight loss interventions: delivery, intensity and effectiveness. *Public health nutrition*, 24(17):5885–5913, 2021.

[72] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[73] Raul I Ramos-Garcia, Eric R Muth, John N Gowdy, and Adam W Hoover. Improving the recognition of eating gestures using intergesture sequential dependencies. *IEEE Journal of Biomedical and Health Informatics*, 19(3):825–831, 2014.

[74] Muhammad Ahmed Raza, Longfei Chen, Li Nanbo, and Robert B Fisher. Eatsense: human centric, action recognition and localization dataset for understanding eating behaviors and quality of motion assessment. *Image and Vision Computing*, 137:104762, 2023.

[75] Philipp V Rouast and Marc TP Adam. Learning deep representations for video-based intake gesture detection. *IEEE Journal of Biomedical and Health Informatics*, 24(6):1727–1737, 2019.

[76] Philipp V Rouast and Marc TP Adam. Single-stage intake gesture detection using ctc loss and extended prefix beam search. *IEEE Journal of Biomedical and Health Informatics*, 25(7):2733–2743, 2021.

[77] Philipp V Rouast, Hamid Heydarian, Marc TP Adam, and Megan E Rollo. Oreba: A dataset for objectively recognizing eating behavior and associated intake. *IEEE Access*, 8:181955–181963, 2020.

[78] James N Salley, Adam W Hoover, Michael L Wilson, and Eric R Muth. Comparison between human and bite-based methods of estimating caloric intake. *Journal of the Academy of Nutrition and Dietetics*, 116(10):1568–1577, 2016.

[79] Dale A Schoeller and David B Allison. Use of doubly-labeled water measured energy expenditure as a biomarker of self-reported energy intake. In *Advances in the Assessment of Dietary Intake*, pages 185–197. CRC Press, 2017.

[80] Jenna L Scisco, Eric R Muth, and Adam W Hoover. Examining the utility of a bite-count–based measure of eating activity in free-living human beings. *Journal of the Academy of Nutrition and Dietetics*, 114(3):464–469, 2014.

[81] Nur Asmiza Selamat and Sawal Hamid Md Ali. Automatic food intake monitoring based on chewing activity: A survey. *IEEE Access*, 8:48846–48869, 2020.

[82] Hogeon Seo, Seunghyeok Back, Seongju Lee, Deokhwan Park, Tae Kim, and Kyoobin Lee. Intra-and inter-epoch temporal context network (iitnet) using sub-epoch features for automatic sleep scoring on raw single-channel eeg. *Biomedical signal processing and control*, 61:102037, 2020.

[83] Mohit Sharma. *Recognition of Eating Activity in Free-Living Conditions Using Wrist-Mounted Inertial Sensing*. PhD thesis, Clemson University, 2019.

[84] Surya Sharma and Adam Hoover. The challenge of metrics in automated dietary monitoring as analysis transitions from small data to big data. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2647–2653. IEEE, 2020.

[85] Surya Sharma and Adam Hoover. Top-down detection of eating episodes by analyzing large windows of wrist motion using a convolutional neural network. *MDPI Bioengineering*, 1(4), 2022.

[86] Surya Sharma, Phillip Jasper, Eric Muth, and Adam Hoover. The impact of walking and resting on wrist motion for automated detection of meals. *ACM Transactions on Computing for Healthcare*, 1(4):1–19, 2020.

[87] Yiru Shen. *Using contextual information to improve hidden Markov model recognition of wrist motions during eating activities*. PhD thesis, Clemson University, 2018.

[88] Yiru Shen, James Salley, Eric Muth, and Adam Hoover. Assessing the accuracy of a wrist motion tracking method for counting bites across demographic and food variables. *IEEE Journal of Biomedical and Health Informatics*, 21(3):599–606, 2016.

[89] Nina Shvetsova, Bart Bakker, Irina Fedulova, Heinrich Schulz, and Dmitry V Dylov. Anomaly detection in medical imaging with deep perceptual autoencoders. *IEEE Access*, 9:118571–118583, 2021.

[90] Theresa A Spiegel, Joel M Kaplan, Antonina Tomassini, and Eliot Stellar. Bite size, ingestion rate, and meal size in lean and obese women. *Appetite*, 21(2):131–145, 1993.

[91] Donna Spruijt-Metz and Wendy Nilsen. Dynamic models of behavior for just-in-time adaptive interventions. *IEEE Pervasive Computing*, 13(3):13–17, 2014.

[92] G. S. D. M. A. Sreenu and Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1):1–27, 2019.

[93] Mingui Sun, Lora E Burke, Zhi-Hong Mao, Yiran Chen, Hsin-Chen Chen, Yicheng Bai, Yuecheng Li, Chengliu Li, and Wenyan Jia. ebutton: a wearable computer for health monitoring and personal assistance. In *Proceedings of the 51st Annual Design Automation Conference*, pages 1–6, 2014.

[94] Zehua Sun, Qiuhong Ke, Hossein Rahmani, Mohammed Bennamoun, Gang Wang, and Jun Liu. Human action recognition from various data modalities: A review. *IEEE transactions on pattern analysis and machine intelligence*, 2022.

[95] Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. Deepsleepnet: A model for automatic sleep stage scoring based on raw single-channel eeg. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, 2017.

[96] Yiping Tang, Yang Zheng, Chen Wei, Kaitai Guo, Haihong Hu, and Jimin Liang. Video representation learning for temporal action detection using global-local attention. *Pattern Recognition*, 134:109135, 2023.

[97] Zeyu Tang and Adam Hoover. A new video dataset for recognizing intake gestures in a cafeteria setting. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 4399–4405. IEEE, 2022.

[98] Zeyu Tang and Adam Hoover. Video-based intake gesture recognition using meal-length context. *ACM Transactions on Computing for Healthcare*, 6(2):1–24, 2025.

[99] Zeyu Tang, Adam Patyk, James Jolly, Stephanie P Goldstein, J Graham Thomas, and Adam Hoover. Detecting eating episodes from wrist motion using daily pattern analysis. *IEEE Journal of Biomedical and Health Informatics*, 28(2):1054–1065, 2023.

[100] Edison Thomaz, Irfan Essa, and Gregory D Abowd. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 1029–1040, 2015.

[101] Frances E Thompson and Amy F Subar. Dietary assessment methodology. *Nutrition in the Prevention and Treatment of Disease*, 1(4):5–48, 2017.

[102] Frances E Thompson, Amy F Subar, Catherine M Loria, Jill L Reedy, and Tom Baranowski. Need for technological innovation in dietary assessment. *Journal of the American Dietetic Association*, 110(1):48, 2010.

[103] Zhigang Tu, Hongyan Li, Dejun Zhang, Justin Dauwels, Baoxin Li, and Junsong Yuan. Action-stage emphasized spatiotemporal vlad for video action recognition. *IEEE Transactions on Image Processing*, 28(6):2799–2812, 2019.

[104] Michele Tufano, Marlou Lasschuijt, Aneesh Chauhan, Edith JM Feskens, and Guido Camps. Capturing eating behavior from video analysis: a systematic review. *Nutrients*, 14(22):4847, 2022.

[105] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[106] Satvik Venkatesh, David Moffat, and Eduardo Reck Miranda. Investigating the effects of training set synthesis for audio segmentation of radio broadcast. *Electronics*, 10(7):827, 2021.

[107] Tri Vu, Feng Lin, Nabil Alshurafa, and Wenyao Xu. Wearable food intake monitoring technologies: A comprehensive review. *Computers*, 6(1):4, 2017.

[108] Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. Deep learning for sensor-based activity recognition: A survey. *Pattern recognition letters*, 119:3–11, 2019.

[109] Youfa Wang, May A Beydoun, Jungwon Min, Hong Xue, Leonard A Kaminsky, and Lawrence J Cheskin. Has the prevalence of overweight, obesity and central obesity levelled off in the united states? trends, patterns, disparities, and future projections for the obesity epidemic. *International Journal of Epidemiology*, 49(3):810–823, 2020.

[110] Haixu Wu, Yao Xu, Jingyuan Wang, Guodong Long, Chengqi Jiang, Tong Zhang, and Linlin Yao. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

[111] Santosh Kumar Yadav, Kamlesh Tiwari, Hari Mohan Pandey, and Shaik Ali Akbar. A review of multimodal human activity recognition with special emphasis on classification, applications, challenges and future directions. *Knowledge-Based Systems*, 223:106970, 2021.

[112] Xitong Yang, Haoqi Fan, Lorenzo Torresani, Larry S Davis, and Heng Wang. Beyond short clips: End-to-end video-level learning with collaborative memories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7567–7576, 2021.

[113] Yafeng Yin, Lei Xie, Zhiwei Jiang, Fu Xiao, Jiannong Cao, and Sanglu Lu. A systematic review of human activity recognition based on mobile devices: overview, progress and trends. *IEEE Communications Surveys & Tutorials*, 26(2):890–929, 2024.

[114] Rui Zhang and Oliver Amft. Bite glasses: measuring chewing using emg and bone vibration in smart eyeglasses. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pages 50–52, 2016.

[115] Rui Zhang and Oliver Amft. Monitoring chewing and eating in free-living using smart eyeglasses. *IEEE Journal of Biomedical and Health Informatics*, 22(1):23–32, 2017.