Single-stage intake gesture detection using CTC loss and extended prefix beam search

Philipp V. Rouast, *Student Member, IEEE,* Marc T. P. Adam

***Abstract*—Accurate detection of individual intake gestures is a key step towards automatic dietary monitoring. Both inertial sensor data of wrist movements and video data depicting the upper body have been used for this purpose. The most advanced approaches to date use a two-stage approach, in which (i) frame-level intake probabilities are learned from the sensor data using a deep neural network, and then (ii) sparse intake events are detected by finding the maxima of the frame-level probabilities. In this study, we propose a single-stage approach which directly decodes the probabilities learned from sensor data into sparse intake detections. This is achieved by weakly supervised training using Connectionist Temporal Classification (CTC) loss, and decoding using a novel extended prefix beam search decoding algorithm. Benefits of this approach include (i) end-to-end training for detections, (ii) consistency with the fuzzy nature of intake gestures, and (iii) avoidance of hard-coded rules. Across two separate datasets, we quantify these benefits by showing relative F_1 score improvements between 2.0% and 6.2% over the two-stage approach for intake detection and eating vs. drinking recognition tasks, for both video and inertial sensors.**

***Index Terms*—Deep learning, CTC, intake gesture detection, dietary monitoring, inertial and video sensors**

I. Introduction

ACCURATE information on dietary intake forms the basis of assessing a person's diet and delivering dietary interventions. To date, such information is typically sourced through memory recall or manual input, for example via dietitians [1] or smartphone apps used to log meals. Such methods are known to require substantial time and manual effort, and are subject to human error [2]. Hence, recent research has investigated how dietary monitoring can be partially automated using sensor data and machine learning [3].

Detection of individual intake gestures in particular is a key step towards automatic dietary monitoring. Wrist-worn inertial sensors provide an unobtrusive way to recognize these gestures. Early work on the Clemson dataset, established in 2012, used threshold values for detection from inertial data [4]. More recent developments include the use of machine learning to learn features automatically [5] and learning from video, which has become more practical with emerging spherical camera technology [6] [7]. Research on the OREBA dataset showed that frontal video data can exhibit even higher accuracies in detecting eating gestures than inertial data [8].

The two-stage approach introduced by Kyritsis et al. [9] is currently the most advanced approach benchmarked on publicly available datasets for both inertial [9] and video data [6]. It first estimates frame-level intake probabilities using deep learning, which are then searched for maxima to detect intake events. Drawbacks of this approach include the explicit nature of the constraint imposed in the second stage, and the loss function not being directly aligned with the detection task.

The authors are with the School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW 2308, Australia. E-mail: philipp.rouast@uon.edu.au, marc.adam@newcastle.edu.au.

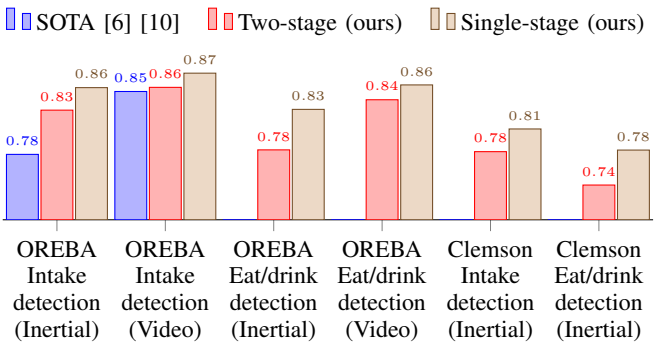

Fig. 1. F_1 scores for our two-stage and single-stage models in comparison with the current state of the art (SOTA). Our single-stage models see relative improvements of 10.2% and 2.6% over the SOTA for inertial [10] and video-based intake detection [6] on the OREBA dataset, and relative improvements between 2.0% and 6.2% over comparable two-stage models for intake detection and eating vs. drinking detection tasks across the OREBA and Clemson datasets.

In this paper, we propose a single-stage approach which directly decodes the probabilities learned from sensor data into sparse intake event detections. This approach is compatible with data from any sensor, including inertial and video. We achieve this by weakly supervised training [11] of the underlying deep neural network with Connectionist Temporal Classification (CTC) loss, and decoding the probabilities using a novel extended prefix beam search algorithm. Compared to the approaches currently established in the literature, our study makes four key contributions:

1) **Single-stage approach.** This is the first study that applies a single-stage approach allowing for end-to-end training with a loss function that directly addresses the intake gesture detection task. We avoid the constraint associated with the second stage of two-stage models [9] [6] (i.e., the two second gap between intake events).
2) **Simplified labels.** The proposed approach requires information about occurrence and order of intake gestures, but not their exact timing. Hence, it is particularly suitable for intake gestures, whose start and end times are fuzzy in nature and highly time-consuming to determine.
3) **Improved performance.** Our single-stage models outperform two-stage models on the OREBA and Clemson datasets, including the current state of the art (SOTA) [6]

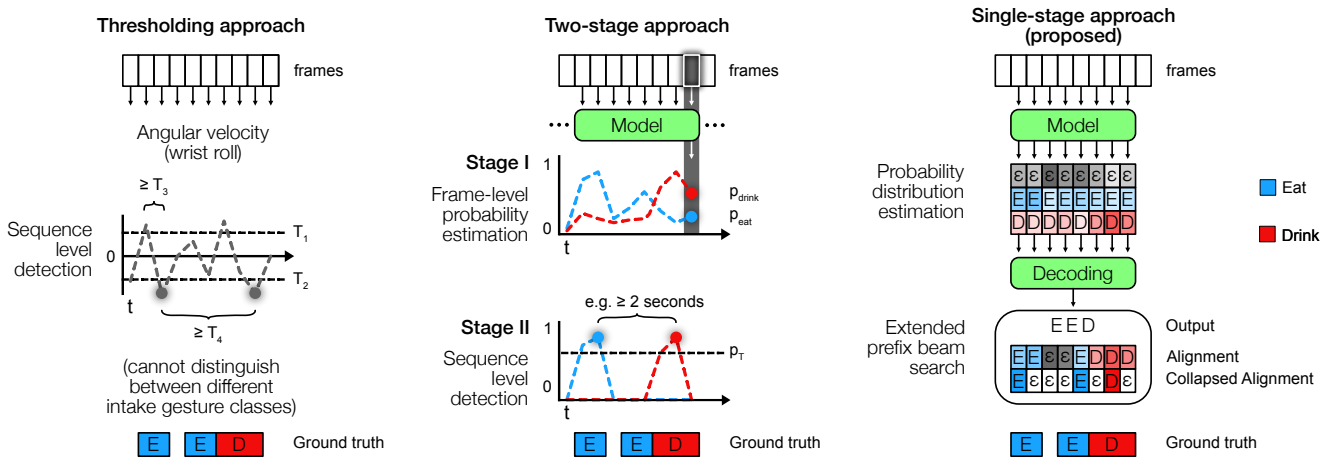

Fig. 2. Comparing existing approaches (left, center) to the proposed approach (right): The thresholding approach [4] (left) searches the angular velocity for values that breach the thresholds T_1 and T_2. The two-stage approach [9] (center) independently estimates frame-level probabilities, which are then searched for maxima on the video level (generalized to two gesture classes here). The proposed single-stage approach (right) directly decodes the estimated probability distribution $p(c|x_t)$ using extended prefix beam search, after which token sequences in the most probable alignment $\hat{A}$ are collapsed to yield the result.

[10] and two-stage versions of our models, see Fig. 1.

4) **Intake gesture recognition.** This is the first study simultaneously detecting and recognizing intake gestures as either eating or drinking from inertial and video data. Distinguishing between eating and drinking is an important step toward more fine-grained analysis of dietary intake.

The remainder of the paper is organized as follows: In Section II, we discuss the related literature on CTC and intake gesture detection. Our proposed method is introduced in Section III, including a complete pseudo-code listing of our proposed decoding algorithm. We present and analyse the evaluation of our proposed model and the SOTA on two datasets in Section IV. Finally, we conclude in Section V.

II. Related Research

A. Intake gesture detection

Intake gesture detection involves the detection of the timestamps at which a person moved their hands to ingest food or drink during an eating occasion. It is one of the three elements of automatic dietary monitoring, which also encompasses recognition of the consumed type of food, and estimation of the consumed quantity of food. Sensors that carry a signal appropriate for the detection of intake gestures include inertial sensors mounted to the wrist [12] and video recordings [6]. Note that information on eating events can also be derived from chewing and swallowing monitored using audio [13] [14], electromyography [15] [16], and piezoelectric sensors [17]. There are also other recent video-based approaches based on skeletal and mouth [18] as well as food, hand and face [7] features extracted using deep learning. For inertial data, there is recent work on in-the-wild monitoring [19]. In the following, we focus on two main approaches for inertial and video data that have been benchmarked on publicly available datasets:

1) Thresholding approach: In 2012, Dong et al. [4] noticed that intake gestures are strongly correlated with the angular velocity around the axis parallel to the wrist (wrist roll). They devised an easily interpretable thresholding approach which requires the angular velocity to first surpass a positive threshold (e.g., rolling wrist one way to pick up food), and then a negative threshold (e.g., rolling wrist the other way to pass food to the mouth). Refer to Fig. 2 (left) for an illustration. The approach selects these thresholds and two further parameters for minimum time amounts during and after a detection based on an exhaustive search of the parameter space. Note that this approach is not generalizable to multiple gesture classes.

2) Two-stage approach: Kyritsis et al. [9] proposed a two-stage approach for detecting intake gestures from accelerometer and gyroscope data. Rouast and Adam [6] later adopted this approach for video data. In this approach, the first stage produces frame-level estimates for the probability of intake versus non-intake. These estimates are provided iteratively by a neural network trained on a sliding two-second context. The second stage identifies the sparse video-level intake gesture timings by operating a thresholded maximum search on the frame-level estimates, constrained by a minimum distance of two seconds between detections. Fig. 2 (center) illustrates this approach generalized to two intake gesture classes.

While this approach is also relatively easy to interpret and works well in practice [19], it has a few restrictions. Firstly, the second stage introduces the explicit constraint of a predefined gap between subsequent intake gestures. This constraint implies that any consecutive events occurring within two seconds of each other lead to false negatives. Secondly, the loss function during neural network training is geared towards optimizing the frame-level predictions, not the video-level detections. In the present work, we address these restrictions by introducing a new single-stage training and decoding approach using CTC – see Fig. 2 (right).

B. Connectionist temporal classification

In 2006, Graves et al. [20] proposed connectionist temporal classification (CTC) to allow direct use of unsegmented input data in sequence learning tasks with recurrent neural networks (RNNs). By interpreting network output as a probability

distribution over all possible token sequences, they derived CTC loss, which can be used to train the network via back-propagation [21]. Hence, what sets CTC apart from previous approaches is the ability to label entire sequences, as opposed to producing labels independently in a frame-by-frame fashion.

While the original application of CTC was phoneme recognition [20], researchers have applied it in various sequence learning tasks such as end-to-end speech recognition [22], handwriting recognition [23], and lipreading [24]. In the most closely related prior research to the present work, Huang et al. [11] extended the CTC framework to enable weakly supervised learning of actions from video, simplifying the required labelling process. To the best of our knowledge, CTC has neither been applied for temporal localization of actions from sensor data nor intake gesture detection.

III. Proposed Method

Our proposed approach interprets the problem of intake gesture detection as a sequence labelling problem using CTC. This allows us to operate within a *single-stage* approach, meaning that both probability estimation and intake gesture detection are operationalized for a single time window of data, as exemplified in Fig. 3:

- The probability distribution $p(c|x_t)$ over all possible token sequences is estimated using a neural network trained with *CTC loss*.
- We decode $p(c|x_t)$ to determine an alignment A using *extended prefix beam search*. We then derive the gesture timings by collapsing event token sequences within A.

The proposed extended prefix beam search is a complex algorithm. To lay the necessary groundwork, we start by introducing the concept of alignments and derive the CTC loss function. Then, we continue by describing greedy decoding and prefix beam search as alternative decoding algorithms which provide the motivation for our extension. We then finally introduce the proposed extended prefix beam search.

A. Alignment between sensor data and labels

In many pattern recognition tasks involving the mapping of input sequences X to corresponding output sequences Y, we encounter problems relating to the alignment between the elements of X and Y. Often, real-world sensor data cannot naturally be aligned with fixed-size tokens: In handwriting recognition, for example, some written letters in X are spatially wider than others, unlike the fixed-size tokens in Y [23]. We face the same problem in intake gesture recognition, where gesture events can have various durations.

To account for the dynamic size of events in the input, we create an alignment A by using the token in question multiple times [25], such as in the example in Fig. 3. The blank token ϵ is additionally introduced to allow separation of multiple instances of the same event class, $A = [E, E, \epsilon, E, E, D, D, D]$ in the example. We derive the token sequence Y from an alignment A by first collapsing repeated tokens and then removing the blank token. Hence, the token sequence for the example is $Y = [E, E, D]$, which correctly reflects the ground truth label. Note that a collapsed output token sequence Y can have many possible corresponding alignments A.

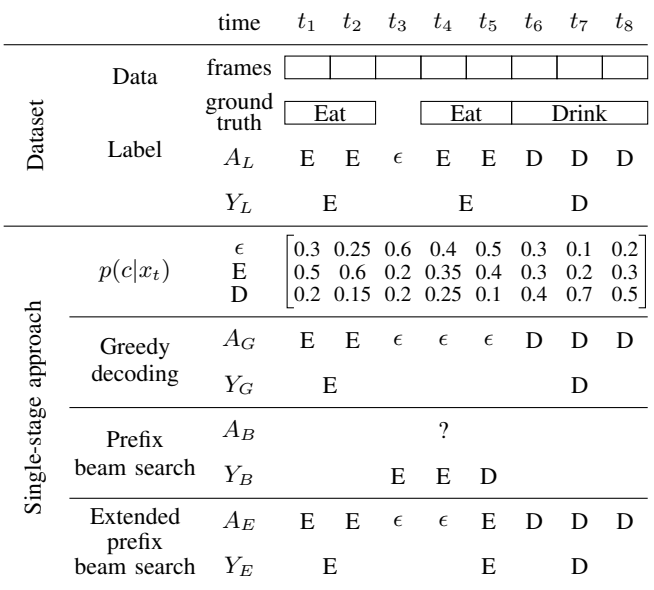

Fig. 3. An example with (1) dataset represented by data and label with corresponding alignment A_L and collapsed token sequence Y_L, (2) the single stage approach for intake gesture detection with estimated probabilities $p(c|x_t)$, and alignments as well as collapsed token sequences produced by *Greedy decoding*, *prefix beam search* as well as *extended prefix beam search*. Note that finding the alignment A_E produced by *extended prefix beam search* is the key element missing for simple *prefix beam search*.

B. CTC loss for probability distribution estimation

Suppose we have an input sequence X of length T, the corresponding output token sequence Y, and possible tokens Σ. Our network is designed to express a probability estimate $p(c|x_t)$ for each token c in Σ given the sensor input x_t at time t. Fig 3 continues the previous example to show what the network output $p(c|x_t)$ might look like. The objective of CTC loss is to minimize the negative log-likelihood of $p(Y|X)$, which is the probability that the network predicts Y when presented with X [20]. This probability can be efficiently computed using dynamic programming, adding the probabilities of the alignments $A_{X,Y}$ that produce Y [21].

$$p(Y|X) = \sum_{A \in A_{X,Y}} \prod_{t=1}^{T} p(c = a_t|x_t) \tag{1}$$

Using CTC loss for intake gesture detection allows our networks to be trained in a weakly supervised fashion with the less restrictive collapsed labels. This implies that our networks will learn to make predictions differently than when trained with cross-entropy loss, as we explore further in Section IV-E. It also implies that examples are required to regularly contain multiple intake gestures for the network to learn properly (e.g., two eating and one drinking gesture in Fig. 3).

C. Greedy decoding

During inference, we decode the probabilities $p(c|x_t)$ into a sequence of tokens Y. This can be interpreted as choosing an alignment A, which is then collapsed to Y. A fast and simple solution is *Greedy decoding*, which chooses the alignment by

selecting the maximum probability token at each time step t [25] as in Equation 2.

$$a_t = \arg\max_{\Sigma} p(c|x_t) \tag{2}$$

However, this method is not guaranteed to produce the most probable Y, since it does not take into account that each Y can have many possible alignments [25]. In the example of Fig 3, greedy decoding gives the alignment $[E, E, \epsilon, \epsilon, \epsilon, D, D, D]$ which collapses to $[E, D]$. Using Equation 1, we can compute that this is indeed an inferior solution to $[E, E, D]$.[1]

D. Prefix beam search

Traversing all possible alignments turns out to be infeasible due to their large number [25]. The *prefix beam search* algorithm [20] uses dynamic programming to search for a token sequence $\hat{Y}$ that maximises $p(\hat{Y}|X)$. It presents a trade-off between computation and solution quality, which can be adjusted through the beam width k, determining how many possible solutions are remembered. Prefix beam search with a beam width of 1 is equivalent to greedy decoding. However, it is important to note that prefix beam search does not remember specific alignments. Hence, it is not possible to temporally localize intake events (see missing A_B in Fig. 3).

The algorithm determines beams in terms of *prefixes* ℓ (candidates for the output token sequence $\hat{Y}$ up to time t), which are stored in a list Y. Each prefix is associated with two probabilities, the first of ending in a blank, $p_b(\ell|x_{1:t})$, and the second of not ending in a blank, $p_{nb}(\ell|x_{1:t})$. For each time step t, the algorithm updates the probabilities for every prefix in Y for the different cases of (i) adding a repeated token and (ii) adding a blank, and adds possible new prefixes. Due to the algorithm design, branches with equal prefixes are dynamically merged. The algorithm then keeps the k best updated prefixes.

E. Extended prefix beam search

Standard prefix beam search finds a token sequence $\hat{Y}$, without retaining information about the alignments $A_{X,\hat{Y}}$. In order to be able to infer the timing of the decoded events in a way consistent with CTC loss, we would like to find $\hat{A}$. This is the most probable alignment that could have produced $\hat{Y}$, as expressed by Equation 3.

$$\hat{A} = \arg\max_{A_{X,\hat{Y}}} \prod_{t=1}^{T} p(c = a_t|x_t) \tag{3}$$

Instead of running a separate algorithm based on $\hat{Y}$, we search for $\hat{A}$ simultaneously as part of prefix beam search, which already includes most of the necessary computation. We add two additional lists for each beam ℓ, $A_b(\ell)$ and $A_{nb}(\ell)$, which store alignment candidates that resolve to ℓ as well as their corresponding probabilities. Every time a probability is updated in prefix beam search, we add new alignment candidates and associated probabilities to the appropriate lists. This includes (i) adding a repeated token, (ii) adding a blank token, and (iii) adding a token that extends the prefix. The algorithm design implies that if two beams with identical prefixes are merged, alignment candidates are also merged dynamically. At the end of each time step t, we resolve the alignment candidates for each ℓ in Y by choosing the highest probability for each $A_b(\ell)$ and $A_{nb}(\ell)$. Finally, for each of the k best token sequences in Y, the best alignment candidate $\hat{A}$ is chosen as the more probable one out of $A_b(\ell)$ and $A_{nb}(\ell)$.

We created a Python implementation[2] of the version listed in Algorithm 1. Note that this version is not created with efficiency in mind. For our experiments, we implemented a more efficient implementation[3] as a C++ TensorFlow kernel.

[1]Meaning that $p([E, D]|X) \approx 0.0719 < 0.1305 \approx p([E, E, D]|X)$

TABLE I
ARCHITECTURES FOR OUR SINGLE-STAGE AND TWO-STAGE MODELS

Layer	Video ResNet-50 CNN-LSTM OREBA		Inertial ResNet-10 CNN-LSTM OREBA		Inertial ResNet-10 CNN-LSTM Clemson
	params	output size	params	output size	output size
data		$16 \times 128^2 \times 3$		512×12	120×6
conv1	$5^2, 64$ stride 1^2	$16 \times 128^2 \times 64$	$1, 64$ stride 1	512×64	120×64
pool1	2^2 stride 2^2	$16 \times 64^2 \times 64$			
conv2	$\begin{bmatrix} 1^2, 64 \\ 3^2, 64 \\ 1^2, 256 \end{bmatrix} \times 3$	$16 \times 64^2 \times 256$	$\begin{bmatrix} 3, 64 \\ 3, 64 \end{bmatrix}$	512×64	120×64
conv3	$\begin{bmatrix} 1^2, 128 \\ 3^2, 128 \\ 1^2, 512 \end{bmatrix} \times 4$	$16 \times 32^2 \times 512$	$\begin{bmatrix} 3, 128 \\ 3, 128 \end{bmatrix}$	256×128	120×128
conv4	$\begin{bmatrix} 1^2, 256 \\ 3^2, 256 \\ 1^2, 1024 \end{bmatrix} \times 6$	$16 \times 16^2 \times 1024$	$\begin{bmatrix} 5, 256 \\ 5, 256 \end{bmatrix}$	128×256	60×256
conv5	$\begin{bmatrix} 1^2, 512 \\ 3^2, 512 \\ 1^2, 2048 \end{bmatrix} \times 3$	$16 \times 8^2 \times 2048$	$\begin{bmatrix} 5, 512 \\ 5, 512 \end{bmatrix}$	64×512	60×512
spatial pool		16×2048			
lstm		16×128		64×64	60×64
dense[a]		$16 \times \lvert\Sigma\rvert$		$64 \times \lvert\Sigma\rvert$	$60 \times \lvert\Sigma\rvert$

[a] Σ includes the blank token, hence $|\Sigma| = 2$ for generic intake gesture detection and $|\Sigma| = 3$ for detection of eating and drinking gestures.

F. Network architectures

Although they are trained with different loss functions, both the single-stage and two-stage approaches each rely on an underlying deep neural network which estimates probabilities. Here, that is for an 8 second window of sensor data from the OREBA and Clemson datasets. We choose adapted versions of the ResNet architecture [26]. Our video network is a CNN-LSTM with a ResNet-50 backbone adjusted for our video resolution. For inertial data, we use a CNN-LSTM with a ResNet-10 backbone using 1D convolutions. Table I reports the parameters and output sizes for all layers.

[2]See https://gist.github.com/prouast/a73354a7586cc6bc444d2013001616b7
[3]Available at https://github.com/prouast/ctc-beam-search-op

Algorithm 1: Extended prefix beam search algorithm (loosely based on [27]): The algorithm stores current prefixes in Y. Probabilities are stored and updated in terms of prefixes ending in blank $p_b(\ell|x_t)$ and non-blank $p_{nb}(\ell|x_t)$, facilitating dynamic merging of beams with identical prefixes. The empty set is used to initialize Y and associated with probability 1 for blank, and 0 for non-blank. $A_b(\ell)$ and $A_{nb}(\ell)$ store the current candidates for alignments (ending in blank and non-blank) pertaining to prefix ℓ, along with their probabilities. They are likewise initialized for the empty prefix. The algorithm then loops over the time steps, updating the prefixes and associated alignments. Each current candidate ℓ is re-entered into the new prefixes Y', adjusting the probabilities for repeated tokens and added blanks. The corresponding alignment candidates and their probabilities are added to the new alignment candidates $A'_{nb}(\ell)$ and $A'_b(\ell)$. Furthermore, for each non-blank token in Σ, a new prefix is created by concatenation, the probability is updated, and corresponding alignment candidates are added. At the end of each time step, we set Y to the k most probable prefixes in Y' and resolve the alignment candidates for each of those prefixes as the most probable ones. Finally, for each of the k best token sequences in Y, the best alignment candidate is chosen as the more probable one out of $A_b(\ell)$ and $A_{nb}(\ell)$.

Data: Probability distributions $p(c|x_t)$ for tokens $c \in \Sigma$ in sensor data x_t from $t = 1, \dots, T$.
Result: k best decoded sequences of tokens Y and best corresponding alignments A.

$p_b(\emptyset|x_{1:0}) \leftarrow 1$, $p_{nb}(\emptyset|x_{1:0}) \leftarrow 0$
$Y \leftarrow \{\emptyset\}$
$A_b(\emptyset) \leftarrow \{(\emptyset, 1)\}$, $A_{nb}(\emptyset) \leftarrow \{(\emptyset, 0)\}$
for $t = 1, \dots, T$ **do**
  $Y' \leftarrow \{\}$
  $A'_b(\cdot) \leftarrow \{\}$, $A'_{nb}(\cdot) \leftarrow \{\}$
  for ℓ **in** Y **do**
    if $\ell \notin Y'$ **then**
      add ℓ to Y'
    end
    if $\ell \neq \emptyset$ **then**
      $p_{nb}(\ell|x_{1:t}) \leftarrow p_{nb}(\ell|x_{1:t}) + p_{nb}(\ell|x_{1:t-1})p(\ell_{|\ell|}|x_{1:t})$
      add (concatenate $A_{nb}(\ell)$ and $\ell_{|\ell|}$, $p_{nb}(\ell|x_{1:t-1})p(\ell_{|\ell|}|x_{1:t})$) to $A'_{nb}(\ell)$
    end
    $p_b(\ell|x_{1:t}) \leftarrow p_b(\ell|x_{1:t}) + p(\epsilon|x_{1:t})(p_b(\ell|x_{1:t-1}) + p_{nb}(\ell|x_{1:t}))$
    add (concatenate $A_b(\ell)$ and ϵ, $p_b(\ell|x_{1:t-1})p(\epsilon|x_{1:t})$) to $A'_b(\ell)$
    add (concatenate $A_{nb}(\ell)$ and ϵ, $p_{nb}(\ell|x_{1:t-1})p(\epsilon|x_{1:t})$) to $A'_b(\ell)$
    for c **in** $\Sigma \setminus \epsilon$ **do**
      $\ell^+ \leftarrow$ concatenate ℓ and c
      add ℓ^+ to Y'
      if $\ell \neq \emptyset$ **and** $c = \ell_{|\ell|}$ **then**
        $p_{nb}(\ell^+|x_{1:t}) \leftarrow p_{nb}(\ell^+|x_{1:t}) + p_b(\ell|x_{1:t-1})p(c|x_{1:t})$
        add (concatenate $A_{nb}(\ell)$ and c, $p_b(\ell|x_{1:t-1})p(c|x_{1:t})$) to $A'_{nb}(\ell^+)$
      else
        $p_{nb}(\ell^+|x_{1:t}) \leftarrow p_{nb}(\ell^+|x_{1:t}) + p(c|x_{1:t})(p_b(\ell|x_{1:t-1}) + p_{nb}(\ell|x_{1:t-1}))$
        add (concatenate $A_b(\ell)$ and c, $p_b(\ell|x_{1:t-1})p(c|x_{1:t})$) to $A'_b(\ell^+)$
        add (concatenate $A_{nb}(\ell)$ and c, $p_{nb}(\ell|x_{1:t-1})p(c|x_{1:t})$) to $A'_{nb}(\ell^+)$
      end
    end
  end
  $Y \leftarrow k$ most probable prefixes in Y'
  for ℓ **in** Y **do**
    $A_b(\ell) \leftarrow$ the most probable sequence in $A'_b(\ell)$
    $A_{nb}(\ell) \leftarrow$ the most probable sequence in $A'_{nb}(\ell)$
  end
end
for ℓ **in** Y **do**
  $A(\ell) \leftarrow$ the most probable sequence in $\{A_b(\ell), A_{nb}(\ell)\}$
end
return Y, A

IV. Experiments and analysis

In the experiments, we compare the proposed single-stage approach to the thresholding approach [4] and the two-stage approach [9] [10]. We consider two datasets of annotated intake gestures: The OREBA dataset [6] and the Clemson Cafeteria dataset [28]. To the best of our knowledge, these are the largest publicly available datasets for intake gesture detection. For both datasets, we attempt detection of generic intake events, as well as simultaneous detection and recognition of eating and drinking gestures. For OREBA, we run separate experiments for inertial and video data. Across our experiments, we use time windows of *8 seconds*, which ensures that examples regularly contain multiple intake events. All code used for the experiments is available at https://github.com/prouast/ctc-intake-detection.

A. Approaches

1) Thresholding approach: We implemented the thresholding approach with four parameters as described by Dong et al. [4] and Shen et al. [28], which only relies on angular velocity (wrist roll). For each dataset, we used the training set to estimate the parameters T_1, T_2, T_3, and T_4.

2) Two-stage approach: SOTA results on OREBA [6] [10] are based on 2 second time windows. However, a 2 second time window is not sufficient for the single-stage approach. Hence, to still facilitate a fair comparison between single-stage and two-stage, we train our own two-stage models based on 8 second time windows and the same architecture as our single-stage models. These models are trained with cross-entropy loss. Video-level detections are reported according to the Stage 2 maximum search algorithm outlined in [9]. To facilitate multi-class comparison, we also extend the Stage 2 search by applying the same threshold to both intake gesture classes.

3) Single-stage approach: Our single-stage models are trained using CTC loss [20]. One caveat of the single-stage approach is that it requires a longer time window than Stage 1 of the two-stage approach. This is to ensure that multiple gestures regularly appear in the training examples, providing a signal for learning of temporal relations. We found that choosing a time window of 8 seconds is just sufficient for this purpose.[4] For inference, the probabilities estimated for each temporal segment are decoded into an alignment using the *Extended prefix beam search* with beam width 10, and then collapsed to yield event detections. On the video level, we first aggregate detections from the individual alignments of sliding windows using frame-wise majority voting before collapse.

B. Training and evaluation metrics

1) Training: All networks are trained using the *Adam* optimizer on the respective training set with batch size 128 for inertial and 16 for video, and an exponentially decreasing learning rate starting at 1e-3. We also use minibatch loss scaling analogous to [6]. Hyperparameter and model selection is based on the validation set unless stated otherwise.

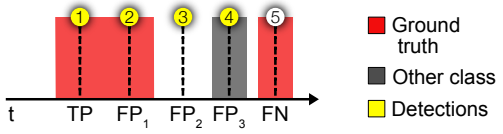

Fig. 4. The evaluation scheme (proposed by [9]; figure from [6] extended here). (1) A true positive is the first detection within each ground truth event; (2) False positives of type 1 are further detections within the same ground truth event; (3) False positives of type 2 are detections outside ground truth events; (4) False positives of type 3 are detections made for the wrong class if applicable; (5) False negatives are non-detected ground truth events.

2) Evaluation: For comparison we use the F_1 measure, applying an extended version of the evaluation scheme proposed by Kyritsis et al. [9] (see Fig. 4). The scheme uses the ground truth to translate sparse detections into measurable metrics for a given label category. As Rouast and Adam [6] report, one correct detection per ground truth event counts as a true positive (TP), while further detections within the same ground truth event are false positives of type 1 (FP_1). Detections outside ground truth events are false positives of type 2 (FP_2) and non-detected ground truth events count as false negatives (FN). We extended the original scheme to support the multi-class case, where detections for a wrong class are false positives of type 3. Based on the aggregate counts, precision ($\frac{TP}{TP+FP_1+FP_2+FP_3}$), recall ($\frac{TP}{TP+FN}$), and the F_1 score ($2 * \frac{Precision*Recall}{Precision+Recall}$) can be calculated.

C. Datasets

1) OREBA: The OREBA dataset [8] includes both inertial and video data. Specifically, we are using the scenario OREBA-DIS with data for 100 participants (69 male, 31 female) and 4790 annotated intake gestures. Data are split into training, validation, and test sets of 61, 20, and 19 participants according to the split suggested by the dataset authors [8]. For our inertial models, we use the processed[5] data from accelerometer and gyroscope readings for both wrists at 64 Hz. The video data comes at a frame rate of 24 fps and spatial resolution of 140x140 pixels. We downsample the video to 2 fps and and use data augmentation analogous to [6], which includes spatial cropping to 128x128 pixels. The choice of 2 fps presents a trade-off as limited GPU memory does not allow us to run experiments based on more than 16 frames at a time. For this dataset, 8 seconds correspond to 16 frames of video at 2 fps and 512 frames of inertial data at 64 Hz.

2) Clemson: The Clemson dataset [28] consists of 488 annotated eating sessions across 264 participants (127 male, 137 female). This results in a combined number of 20644 intake gestures (referred to as *bites* in the original paper). Sensor data for accelerometer and gyroscope is available for the dominant hand at 15 Hz. We split the sessions into training, validation, and test sets (302, 93 and 93 sessions respectively) such that each participant appears in only one of the three. Details are

[4]For time windows of 8 seconds, multiple gestures appear in 8.2% (OREBA) / 7.7% (Clemson) of examples, which means that we can expect 1.3 (OREBA) / 1.2 (Clemson) such gestures on average in batches of size 16.

[5]Processing includes mirroring for data uniformity, removal of the gravity effect using Madgwick's filter [29], and standardization.

TABLE II
RESULTS FOR THE OREBA AND CLEMSON DATASETS (TEST SET)

Method	Dataset	Modality	Generic intake events F_1	(E)ating and (D)rinking F_1^E	F_1^D	$F_1^{E\wedge D}$
Thresholding [4] ($T_1 = 25$, $T_2 = -25$, $T_3 = 2$, $T_4 = 2$, 64 Hz)	OREBA	Inertial	0.275			
Two-stage CNN-LSTM [10] (2 sec @ 64 Hz)[a]	OREBA	Inertial	0.778			
Two-stage ResNet-10 CNN-LSTM (ours, 8 sec @ 64 Hz)	OREBA	Inertial	0.831	0.798	0.638	0.783
Single-stage ResNet-10 CNN-LSTM (ours, 8 sec @ 64 Hz)	OREBA	Inertial	**0.858**	**0.837**	**0.770**	**0.832**
Two-stage ResNet-50 SlowFast [6] (2 sec @ 8 fps)[a]	OREBA	Video	0.853			
Two-stage ResNet-50 CNN-LSTM (ours, 8 sec @ 2 fps)	OREBA	Video	0.858	0.841	**0.859**	0.843
Single-stage ResNet-50 CNN-LSTM (ours, 8 sec @ 2 fps)	OREBA	Video	**0.875**	**0.870**	0.766	**0.861**
Thresholding [4] ($T_1 = 15$, $T_2 = -15$, $T_3 = 1$, $T_4 = 4$, 15 Hz)	Clemson	Inertial	0.362			
Two-stage ResNet-10 CNN-LSTM (ours, 8 sec @ 15 Hz)	Clemson	Inertial	0.781	0.743	0.733	0.741
Single-stage ResNet-10 CNN-LSTM (ours, 8 sec @ 15 Hz)	Clemson	Inertial	**0.808**	**0.773**	**0.863**	**0.783**

[a] SOTA models from [10] and [6]; test set results as reported in [8]. These models use time windows of 2 seconds, while our models require time windows of 8 seconds due to the nature of the single-stage approach.

available in Section S2 of the Supplementary Material. For this dataset, 8 seconds correspond to 120 samples. Before feeding the sensor data into our models, we apply the same preprocessing as for OREBA.

D. Results

Results are listed in Table II, and extended results with detailed metric counts are available in Section S1 of the Supplementary Material.

1) Detecting intake gestures: Here, the goal is to detect only one generic intake event class. The results displayed in the center column of Table II reveal that the single-stage approach generally yields higher performance than the thresholding and two-stage approaches (average improvement of 6.4% over SOTA and 2.9% over two-stage versions of our own models).

For OREBA, the relative improvement over the SOTA equals 10.2% and 2.6% for the inertial and video modalities, respectively. In the same vein, we measure improvements of 3.2% and 2.0% over the two-stage versions of our own models. We can make an observation regarding the difference between inertial and video results on OREBA: For inertial, our two-stage model with 8 second time window leads to a significant improvement over the two-stage SOTA – accounting for ca. 66% of the improvement recorded for our single-stage model over the two-stage SOTA. For video on the other hand, the same figure is only ca. 23%. A plausible explanation for this observation is that a larger time window does not make up for missing detail due to the reduced frame rate.

Besides the thresholding approach [4] [28], we are not aware of any SOTA deep learning models on the Clemson dataset. The results demonstrate that both the two-stage and single-stage approach outperform thresholding by a large margin. This not surprising since thresholding exclusively relies on one channel of gyroscope data and the deep learning models have many more parameters. Comparing our own models, we find that the single-stage approach leads to a relative improvement of 3.5%, which is in a similar ballpark to the results for OREBA. It is worth noting that the F_1 scores are generally lower for the Clemson than for the OREBA, indicating that it is more challenging for intake gesture detection. However, this may be related to the lower sampling rate in Clemson and the fact that data for both wrists is available for OREBA, while only the dominant wrist is included in Clemson.

2) Simultaneous detection of intake events and recognition of eating vs. drinking: This task consists of detection and simultaneous recognition of intake events as either eating or drinking. As there is no current SOTA for this more fine-grained classification, we solely compare results for the separately trained two-stage and single-stage versions of our own models. In the right hand side columns of Table II, we report separate F_1 scores for eating and drinking individually, as well as both together.

We can make three main observations: Firstly, the single-stage approach again outperforms the two-stage approach for both datasets and modalities, however the result is more pronounced for inertial data with an average relative improvement of 5.9%. Secondly, the increased difficulty of this task compared to the generic detection task is noticeable in the difference between the F_1 and $F_1^{E\wedge D}$ scores, a decrease of 3.0% for OREBA and 4.1% for Clemson. Thirdly, there is no clear indication whether eating or drinking is easier to detect. While the average across both datasets and modalities hints at eating being easier, this does not hold true for all combinations.

Additionally, it is interesting to note that there are generally very few misclassifications between eating and drinking. As indicated by Table III, the frequency of false positives of type 2 is higher than the frequency of false positives of type 3 by almost two orders of magnitude.

E. Effect of training with CTC loss or cross-entropy loss

During our introduction of CTC loss in Section III-B, we mentioned that weakly supervised training with CTC causes our networks learn a different approach of detecting events than cross-entropy loss. We can think of cross-entropy loss as causing the network to predict *whether a frame occurs anytime during* the gesture that is being detected. The analogous way of thinking about CTC loss is to predict *which frames are the most distinctive about* the gesture that is being detected. This causes the signature for predictions by our single-stage models to look more like probability spikes, while the two-stage models produce sequences of high probability values.

We illustrate this characteristic difference between the single-stage and two-stage approaches in Fig. 5 using an

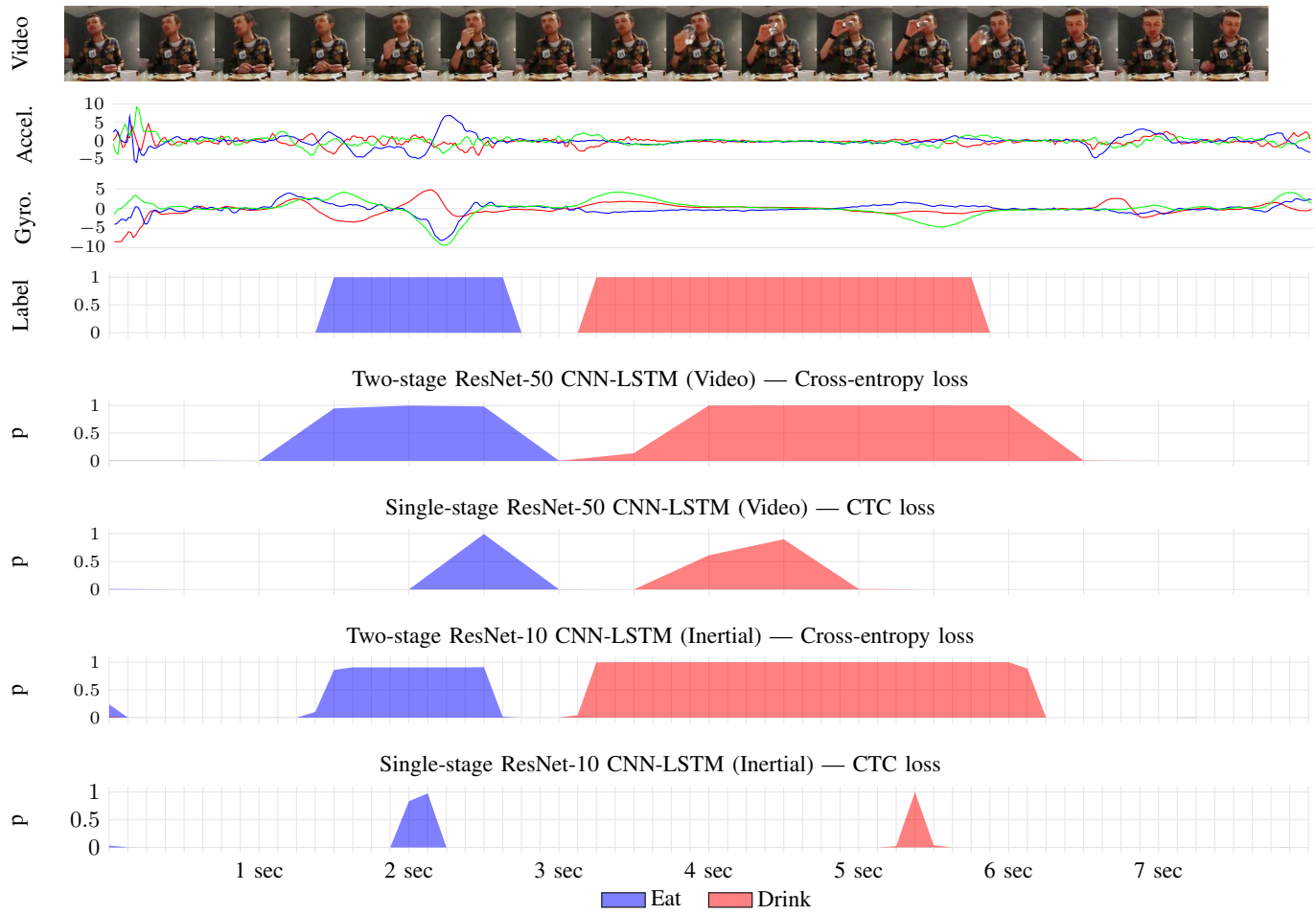

Fig. 5. Illustrating the effect of training with CTC loss or cross-entropy loss using input data, label, and model predictions for one 8 second example from the OREBA validation set.

TABLE III
AVERAGED RESULTS ACROSS ALL EXPERIMENTS (TEST SET). NUMBER OF TP, FP_1, FP_2, FP_3, AND FN ARE EXPRESSED AS PERCENTAGES OF THE RESPECTIVE GROUND TRUTH NUMBER OF GESTURES TO FACILITATE COMPARISONS.

Method	TP [%]	FP_1 [%]	FP_2 [%]	FP_3 [%]	FN [%]	F_1
Two-stage	76.39	2.15	10.80	0.17	23.61	0.8063
Single-stage, greedy decoding	79.58	0.48	**10.60**	0.15	20.42	0.8344
Single-stage, extended prefix beam search	**80.55**	**0.48**	11.60	**0.15**	**19.45**	**0.8361**

example from the validation set of OREBA for eating and drinking detection. Here, time-synchronized 2 fps video and 64 Hz inertial data (dominant hand) for one 8 second time window are plotted alongside the ground truth and predictions of the corresponding two-stage and single-stage models. Note that the output frequency of the models differs, with 2 Hz for the video model and 8 Hz for the inertial models, respectively.

We observe that the predictions by the two-stage models indeed mimick the ground truth, while the single-stage models produce probability spikes where events are detected. Furthermore, these probability spikes line up temporally with the patterns that are most distinct about the gestures for the human eye. That is, the single-stage video model spikes at exactly the frames where the participant begins ingesting the food and drink. For the inertial data it is more difficult to interpret, but the times where the spikes occur are also associated with the most pronounced changes in the inertial signal.

When averaging the results across all datasets and tasks as reported in Table III, it becomes clear that training with CTC loss accounts for the majority of the improvement of single-stage models over two-stage models. The effect of training with CTC loss manifests itself in a higher true positive rate and an associated lower false negative rate. Furthermore, there is a significant drop in false positives of type 1, which were previously conjectured to be a restriction of the two-stage approach [6]. In particular, the single-stage approach avoids the hardcoded 2 second gap in Stage 2 of the two-stage approach and is thus less likely to lead to false positives of type 1 for gestures with a long duration.

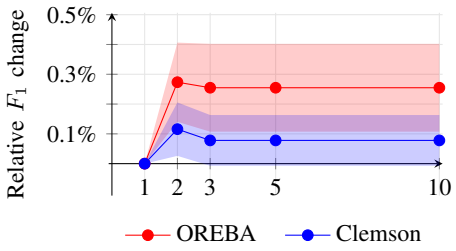

Fig. 6. Average relative F_1 change with standard deviation when choosing different beam widths for decoding our models on the test set. The base scenario is a beam width of 1, which corresponds to *greedy decoding*. We observe that extended prefix beam search decoding mainly benefits the OREBA models, and there are no improvements for beam widths greater than 2.

F. Difference between Greedy decoding and Extended beam search decoding

In theory, the results produced by the proposed extended prefix beam search decoding better reflect the network's intended output than greedy decoding, since they are computed in the same way as CTC loss works internally. However, the reality of our scenario is characterized by few classes and relatively low uncertainty. This is also indicated by the low rate of false positives of type 3 in Table III and also the high prediction confidences in Fig 5. Hence, it turns out that the effect of extended prefix beam search decoding is not very noticeable - a relative improvement of only 0.20% over greedy decoding as indicated by Table III. This improvement is characterized by a higher true positive rate and an associated lower false negative rate, but also a higher rate of false positives of type 2.

Further to this point, recall that extended prefix beam search decoding with a beam width of 1 is equivalent to greedy decoding. While previously reported results are based on beam width 10, experiments with other beam widths show that values over 2 do not lead to further improvements. As illustrated in Fig. 6, extended prefix beam search decoding with beam widths greater than 1 mainly had benefits for the OREBA models.

V. Conclusion

In this paper, we introduced a single-stage approach to detect and simultaneously recognize intake gestures. This is achieved by weakly supervised training of a deep neural network with CTC loss and decoding using a novel extended prefix beam search decoding algorithm. Using CTC loss instead of cross-entropy loss allows us to interpret intake gesture detection as a sequence labelling problem, where the network labels an entire sequence as opposed to doing this independently in a frame-by-frame fashion. Additionally, to the best of our knowledge, we are the first to attempt simultaneous detection of intake gestures and distinction between eating and drinking using deep learning. We demonstrate improvements over the established two-stage approach [9] [6] using two datasets. These improvements apply to both generic intake gesture detection and eating vs. drinking recognition tasks, and also to both video and inertial sensor data.

The proposed extended prefix beam search decoding algorithm is the second novel element in this context besides CTC loss. This algorithm allows us to decode the probability estimate provided by the deep neural network in a way that is consistent with the computation of CTC loss. However, despite the theoretical benefits of this algorithm, our results show that training with CTC loss accounts for the lion's share of the improvements we see over the two-stage approach. This could be explained by to the low number of classes for the datasets and tasks considered here. Greedy decoding can hence be seen as a fast baseline alternative. It remains to be seen in future work whether extended prefix beam search decoding is more useful when working with a larger number of classes and higher associated uncertainty.

Limitations of the single-stage approach include a requirement for a larger time window during training than the two-stage approach. This is required to assure that multiple intake gestures are regularly presented during training, as a basis for learning of the temporal interplay between intake gestures. It follows that the single-stage approach also has a requirement for more GPU memory, since more activations and gradients have to be stored during training. In our work, this mainly had an impact for the video model, which has a large memory footprint to begin with.

This work has several implications for future research. We have shown a feasible way of detecting intake gestures while simultaneously classifying them into eating and drinking. Given larger video datasets with more different food types and associated labels, it should be possible to perform more fine-grained classification of different foods. The necessity of large datasets has been pointed out [30] and detailed food classes are in fact available for the Clemson dataset, but tentative experiments indicated that inertial sensor data may not be sufficiently expressive to yield satisfactory results for food classification. Another implication directly has to do with the practical task of labelling future datasets. When working with CTC loss, events do not need to be painstakingly labelled with a start and end timestamp. Instead, it is sufficient to mark the apex of the gesture – similar to how the single-stage approach makes detections – which has the potential to significantly reduce the labelling workload and reduce ambiguity around determining the exact start and end times of intake gestures.

Acknowledgment

We gratefully acknowledge the support by the Bill & Melinda Gates Foundation [OPP1171389]. This work was additionally supported by an Australian Government Research Training (RTP) Scholarship.

References

[1] G. Block, "A review of validations of dietary assessment methods," *Am. J. Epidemiology*, vol. 115, no. 4, pp. 492–505, 1982.

[2] S. W. Lichtman, K. Pisarska, E. R. Berman, M. Pestone, H. Dowling, E. Offenbacher, H. Weisel, S. Heshka, D. E. Matthews, and S. B. Heymsfield, "Discrepancy between self-reported and actual caloric intake and exercise in obese subjects," *New England J. Medicine*, vol. 327, no. 27, pp. 1893–1898, 1992.

[3] T. Vu, F. Lin, N. Alshurafa, and W. Xu, "Wearable food intake monitoring technologies: A comprehensive review," *Computers*, vol. 6, no. 1, p. 4, 2017.

[4] Y. Dong, A. Hoover, J. Scisco, and E. Muth, "A new method for measuring meal intake in humans via automated wrist motion tracking," *Applied psychophysiology and biofeedback*, vol. 37, no. 3, pp. 205–215, 2012.
[5] K. Kyritsis, C. Diou, and A. Delopoulos, "Food intake detection from inertial sensors using lstm networks," in *Proc. Int. Conf. Image Analysis and Processing*, 2017, pp. 411–418.
[6] P. Rouast and M. Adam, "Learning deep representations for video-based intake gesture detection," *IEEE J. Biomedical and Health Informatics*, vol. 24, no. 6, pp. 1727–1737, 2019.
[7] J. Qiu, F. P.-W. Lo, and B. Lo, "Assessing individual dietary intake in food sharing scenarios with a 360 camera and deep learning," in *Proc. IEEE International Conference on Wearable and Implantable Body Sensor Networks*, 2019.
[8] P. V. Rouast, H. Heydarian, M. T. P. Adam, and M. Rollo, "Oreba: A dataset for objectively recognizing eating behaviour and associated intake," *arXiv preprint arXiv:1611.01599*, 2020.
[9] K. Kyritsis, C. Diou, and A. Delopoulos, "Modeling wrist micromovements to measure in-meal eating behavior from inertial sensor data," *IEEE J. Biomedical and Health Informatics*, 2019.
[10] H. Heydarian, P. V. Rouast, M. T. P. Adam, T. Burrows, and M. E. Rollo, "Deep learning for intake gesture detection from wrist-worn inertial sensors: The effects of preprocessing, sensor modalities, and sensor positions," *Working paper*, 2020.
[11] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling," in *European Conference on Computer Vision*, 2016, pp. 137–153.
[12] H. Heydarian, M. Adam, T. Burrows, C. Collins, and M. E. Rollo, "Assessing eating behaviour using upper limb mounted motion sensors: A systematic review," *Nutrients*, vol. 11, no. 5, p. 1168, 2019.
[13] O. Amft, M. Stager, P. Lukowicz, and G. Troster, "Analysis of chewing sounds for dietary monitoring," in *Proc. UbiComp*, 2005, pp. 56–72.
[14] O. Amft, M. Kusserow, and G. Troster, "Bite weight prediction from acoustic recognition of chewing," *IEEE Trans. Biomedical Eng.*, vol. 56, no. 6, pp. 1663–1672, 2009.
[15] R. Zhang and O. Amft, "Monitoring chewing and eating in free-living using smart eyeglasses," *IEEE J. Biomedical and Health Informatics*, vol. 22, no. 1, pp. 23–32, 2018.
[16] ——, "Retrieval and timing performance of chewing-based eating event detection in wearable sensors," *Sensors*, vol. 20, no. 2, p. 557, 2020.
[17] E. S. Sazonov and J. M. Fontana, "A sensor system for automatic detection of food intake through non-invasive monitoring of chewing," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1340–1348, 2012.
[18] D. Konstantinidis, K. Dimitropoulos, B. Langlet, P. Daras, and I. Ioakimidis, "Validation of a deep learning system for the full automation of bite and meal duration analysis of experimental meal videos," *Nutrients*, vol. 12, no. 1, p. 209, 2020.
[19] K. Kyritsis, C. Diou, and A. Delopoulos, "A data driven end-to-end approach for in-the-wild monitoring of eating behavior using smartwatches," *IEEE J. Biomedical and Health Informatics*, 2020.
[20] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.
[21] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph.D. dissertation, Technische Universität München, 2008.
[22] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.
[23] M. Liwicki, A. Graves, S. Fernàndez, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.
[24] Y. M. Assael, B. Shillingford, S. Whiteson, and N. De Freitas, "Lipnet: End-to-end sentence-level lipreading," *arXiv preprint arXiv:1611.01599*, 2016.
[25] A. Hannun, "Sequence modeling with ctc," *Distill*, 2017.
[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[27] A. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," *arXiv preprint arXiv:1408.2873*, 2014.
[28] Y. Shen, J. Salley, E. Muth, and A. Hoover, "Assessing the accuracy of a wrist motion tracking method for counting bites across demographic and food variables," *IEEE J. Biomedical and Health Informatics*, vol. 21, no. 3, pp. 599–606, 2017.
[29] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," University of Bristol (UK), Tech. Rep., 2010.
[30] Y. Shen, E. Muth, and A. Hoover, "The impact of quantity of training data on recognition of eating gestures," *arXiv preprint arXiv:1812.04513*, 2018.

Philipp V. Rouast received the B.Sc. and M.Sc. degrees in Industrial Engineering from Karlsruhe Institute of Technology, Germany, in 2013 and 2016 respectively. He is currently working towards the Ph.D. degree in Information Systems and is a graduate research assistant at The University of Newcastle, Australia. His research interests include deep learning, affective computing, HCI, and related applications of computer vision. Find him at https://www.rouast.com.

Marc T. P. Adam is an Associate Professor in Computing and Information Technology at the University of Newcastle, Australia. In his research, he investigates the interplay of human users' cognition and affect in human-computer interaction. He is a founding member of the Society for NeuroIS. He received an undergraduate degree in Computer Science from the University of Applied Sciences Würzburg, Germany, and a PhD in Information Systems from Karlsruhe Institute of Technology, Germany.