Single-Stage Intake Gesture Detection Using CTC Loss and Extended Prefix Beam Search

Philipp V. Rouast, *Member, IEEE* and 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) simplified timing requirements for intake gesture labels, and (iii) improved detection performance compared to existing approaches. Across two separate datasets, we achieve relative F_1 score improvements between 1.9% and 6.2% over the two-stage approach for intake detection and eating/drinking detection tasks, for both video and inertial sensors.**

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

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 detect these gestures.

Manuscript received August 4, 2020; revised November 18, 2020; accepted December 18, 2020. Date of publication December 23, 2020; date of current version July 20, 2021. This work was supported in part by Bill and Melinda Gates Foundation under Grant OPP1171389 and in part by Australian Government Research Training (RTP) Scholarship. *(Corresponding author: Marc Adam.)*

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).

This paper has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/JBHI.2020.3046613

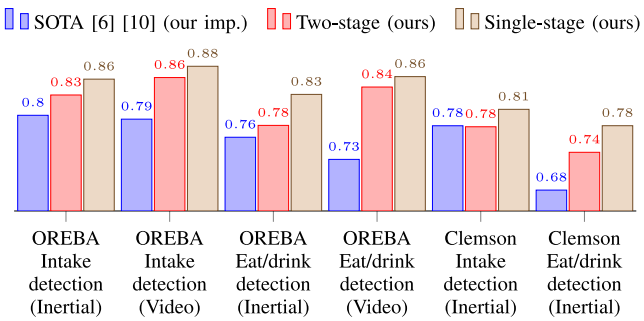

Fig. 1. F_1 scores for our two-stage and single-stage models in comparison with the state of the art (SOTA). Our single-stage models see relative improvements between 3.3% and 17.7% over our implementations of the SOTA for inertial [10] and video modalities [6], and relative improvements between 1.9% and 6.2% over our own two-stage models for intake detection and eating/drinking detection across the OREBA and Clemson datasets.

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. Thereby, the two-stage approach builds on a predefined gap between intake gestures in the second stage.

In this paper, we propose a single-stage approach which directly decodes the probabilities learned from sensor data into sparse intake event detections. 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 existing approaches 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. Thereby, we avoid the predefined gap between subsequent intake gestures in the second stage of two-stage models [9], [6].

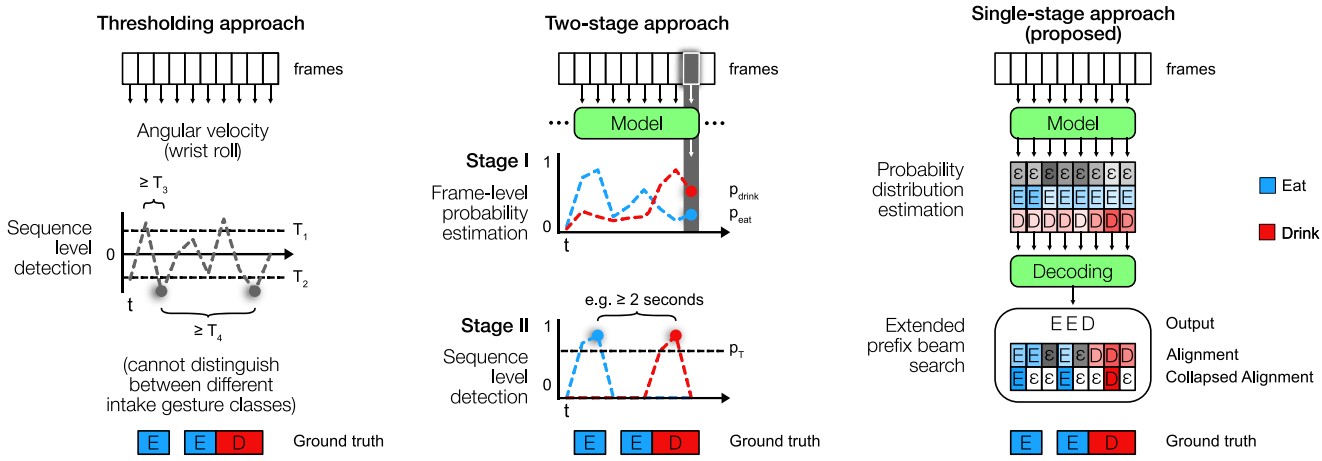

Fig. 2. Comparing existing approaches (left, center) to the proposed approach (right): The thresholding approach [4] (left) searches the angular velocity for values that breach the thresholds T_1 and T_2. The two-stage approach [9] (center) independently estimates frame-level probabilities, which are 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.

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 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], [10] and two-stage versions of our models, see Fig. 1.
4) **Intake gesture detection.** This is the first study to perform simultaneous localization and classification[1] of intake gestures. While we use the example of eating and drinking, the approach could also be applied to more fine-grained analysis of dietary intake given appropriate data.

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 discuss the relative merits of the single-stage and two-stage approaches in Section V and conclude in Section VI.

II. Related Research

A. Intake Gesture Detection

Intake gesture detection involves the determination 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 classification 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] devised an easily interpretable thresholding approach which requires the angular velocity around the wrist to first surpass a positive threshold (e.g., rolling one way to pick up food), and then a negative threshold (e.g., rolling 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], there are a few aspects that need to be considered. Firstly, the second stage requires a predefined gap of two seconds between subsequent intake gestures. This

[1]For the purpose of this study, gesture *detection* refers to temporal localization and simultaneous classification of a gesture (e.g., as a generic intake gesture, or as an eating or drinking gesture).

predefined gap implies that 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 propose an alternative approach 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 backpropagation [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]. Further, CTC has also been applied to sign language recognition from wrist-worn inertial sensor data [25], [26]. 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 this day, 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 inference is operationalized for a single time window of data, as exemplified in Fig. 3:

- A probability distribution over possible events for each time step is estimated using a neural network previously trained with *CTC loss* [20].
- These probabilities are decoded using *extended prefix beam search* and collapsed to derive the gesture timings.

We start by introducing the concept of alignments as well as the CTC loss function. Then, we describe greedy decoding and prefix beam search as alternative decoding algorithms which provide the motivation for our extension. Finally, we 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 challenges relating to the alignment between the elements of X and Y. This is because real-world sensor data cannot always 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]. A similar challenge arises in intake gesture detection, where gesture events can have various durations.

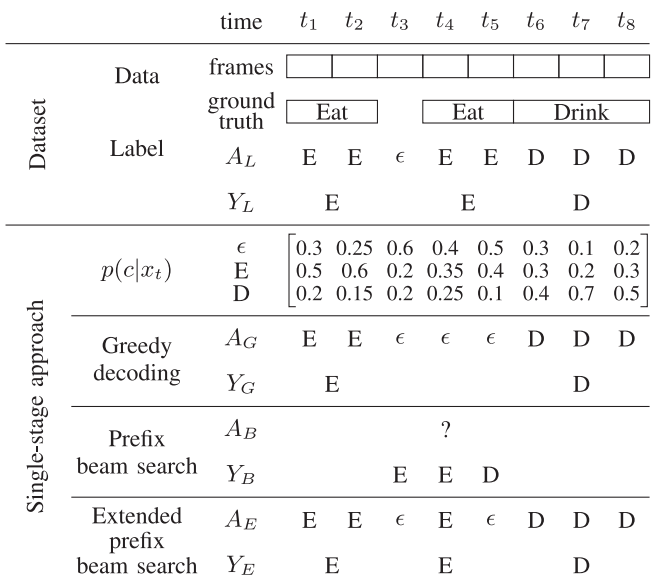

Fig. 3. Example with (1) dataset represented by data and label with corresponding alignment A_L and collapsed token sequence Y_L, (2) the single stage approach for intake gesture detection with estimated probabilities $p(c|x_t)$, and alignments as well as collapsed token sequences produced by *greedy 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*.

To account for the dynamic size of events in the input, we create an alignment A by using the token in question multiple times [27], such as in the example in Fig. 3. In addition, we introduce the blank token ϵ 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. Any one collapsed output token sequence Y can have many possible corresponding alignments A.

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. This probability can be expressed in the form given in Equation (1) [27], building on the individual tokens a_t in all valid alignments $A_{X,Y}$ between X and Y.

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

To train our single-stage networks for intake gesture detection, we use an implementation of CTC loss included in

TensorFlow [28]. This training process can be characterized as weakly supervised, since it only requires the less restrictive collapsed labels Y which do not include timing information besides occurrence and order of the tokens. An implication of using CTC loss is that our networks 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 [27]. 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. 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 $[E, E, D]$.[2]

D. Prefix Beam Search

Traversing all possible alignments turns out to be infeasible due to their large number [27]. 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 the algorithm remembers. 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 infer the *timing* of the decoded events in a way consistent with CTC loss, the goal of our extended prefix beam search is to find $\hat{A}$. This is the most probable alignment that could have produced $\hat{Y}$, as expressed by 2.

$$\hat{A} = \underset{A_{X,\hat{Y}}}{\arg\max} \prod_{t=1}^{T} p(c = a_t|x_t) \tag{2}$$

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[3] of the pseudo-code shown in Algorithm 1. Note that this version is not created with efficiency in mind. For our experiments, we implemented a more efficient implementation[4] as a C++ TensorFlow kernel.

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. We choose adapted versions of the ResNet architecture [29]. 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.

IV. Experiments and Analysis

In the experiments, we compare the proposed single-stage approach to the thresholding [4] and the two-stage approach [9], [10] using two datasets of annotated intake gestures (OREBA [6] and Clemson [31]). To this day, these are the largest publicly available datasets for intake gesture detection. For both datasets, we attempt detection of generic intake gestures, as well as detection of eating and drinking gestures. 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.

[2]Specifically, applying CTC loss to the numerical example in Fig. 3 we find that $p([E, D]|X) \approx 0.0719 < 0.1305 \approx p([E, E, D]|X)$.

[3]See https://gist.github.com/prouast/a73354a7586cc6bc444d2013001616b7

[4]Available at https://github.com/prouast/ctc-beam-search-op

Algorithm 1: Extended prefix beam search algorithm (loosely based on [30]): 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 $t = 1, \ldots, T$.
Result: k best decoded sequences of tokens Y and best corresponding alignments A.

$p_b(\emptyset|x_{1:0}) \leftarrow 1$, $p_{nb}(\emptyset|x_{1:0}) \leftarrow 0$
$Y \leftarrow \{\emptyset\}$
$A_b(\emptyset) \leftarrow \{(\emptyset, 1)\}$, $A_{nb}(\emptyset) \leftarrow \{(\emptyset, 1)\}$
for $t = 1, \ldots, T$ **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 $\ell_{|\ell|}$, $p(A_{nb}(\ell))p(\ell_{|\ell|}|x_{1:t})$) to $A'_{nb}(\ell)$
    end
    $p_b(\ell|x_{1:t}) \leftarrow p_b(\ell|x_{1:t}) + p(\epsilon|x_{1:t})(p_b(\ell|x_{1:t-1}) + p_{nb}(\ell|x_{1:t-t}))$
    add (concatenate $A_b(\ell)$ and ϵ, $p(A_b(\ell))p(\epsilon|x_{1:t})$) to $A'_b(\ell)$
    add (concatenate $A_{nb}(\ell)$ and ϵ, $p(A_{nb}(\ell))p(\epsilon|x_{1:t})$) 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, $p(A_b(\ell))p(c|x_{1:t})$) 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, $p(A_b(\ell))p(c|x_{1:t})$) to $A'_b(\ell^+)$
        add (concatenate $A_{nb}(\ell)$ and c, $p(A_{nb}(\ell))p(c|x_{1:t})$) 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

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
pool		16×2048			
lstm		16×128		64×64	60×64
dense[a]		$16 \times \lvert\Sigma\rvert$		$64 \times \lvert\Sigma\rvert$	$60 \times \lvert\Sigma\rvert$

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

A. Approaches

1) Thresholding Approach: We implemented the thresholding approach with four parameters as described by Dong *et al.* [4] and Shen *et al.* [31], 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 s time windows, which is not sufficient for the single-stage approach. Hence, to facilitate a fair comparison, we also train several two-stage models based on 8 s time windows. In particular, we use cross-entropy loss to train two-stage versions of our own architectures outlined in Table I, as well as the architectures proposed in Heydarian *et al.* [10], Rouast *et al.* [6], and the adapted version of Kyritsis *et al.* [9] used in [10]. Note that the latter was originally designed to be trained with additional sub-gesture labels which are not available for the Clemson and OREBA datasets. These models are trained with cross-entropy loss. Detections on the video level are reported according to the Stage 2 maximum search algorithm by [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 temporal relations. At the same time, due to memory restrictions for the video model, longer time windows come with the drawback of having to reduce the sampling rate of the input data. In light of this tradeoff, we considered different configurations and ultimately decided for a window size of 8 seconds.[5] For inference, the probabilities estimated for each temporal segment are decoded into an alignment using the *Extended prefix beam search*, and then collapsed to yield event detections. Based on an analysis on the validation set (see Section IV-F), we used a beam width of 3. On the video level, we first aggregate detections from the individual alignments of sliding windows using frame-wise majority voting before collapse.

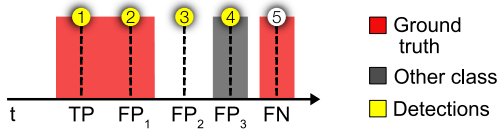

Fig. 4. Evaluation scheme (proposed by [9]; figure adapted from [6]). (1) A true positive is the first detection within each ground truth event; (2) False positives of type 1 are further detections within the same ground truth event; (3) False positives of type 2 are detections outside ground truth events;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

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. We use an exponentially decreasing learning rate starting at 1e-3, except for the SOTA implementations where we use the learning rate settings reported by the authors [10], [9] [6]. We also use minibatch loss scaling, analogously to [6]. Hyperparameter and model selection is based on the validation set.

2) Evaluation: For comparison we use the F_1 measure, applying an extension of the evaluation scheme 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 class. 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 of a wrong class are false positives of type 3 (FP_3). Using the aggregate counts, we calculate precision, recall, and F_1.

C. Datasets

1) OREBA: The OREBA dataset [8] includes inertial and video data. This dataset was approved by the IRB at The University of Newcastle on 10 September 2017 (H-2017-0208).

[5]A window size of 8 seconds allows a video sampling rate of 2 fps and translates into a 74.7% chance of seeing at least one example with multiple gestures per batch during video model training on OREBA. Details on the considered window sizes can be found in the Supplemental Material S1.

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

Method	Dataset	Modality	Intake gestures	(E)ating and (D)rinking gestures		
			F_1	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 CNN-LSTM [9] (our implementation, 8 sec @ 64 Hz)	OREBA	Inertial	0.740	0.732	0.657	0.726
Two-stage CNN-LSTM [10] (our implementation, 8 sec @ 64 Hz)	OREBA	Inertial	0.799	0.772	0.696	0.765
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	**0.855**	**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 SlowFast [6] (our implementation, 8 sec @ 2 fps)	OREBA	Video	0.793	0.751	0.566	0.730
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**	**0.869**	0.761	**0.859**
Thresholding [4] ($T_1 = 15$, $T_2 = -15$, $T_3 = 1$, $T_4 = 4$, 15 Hz)	Clemson	Inertial	0.362			
Two-stage CNN-LSTM [9] (our implementation, 8 sec @ 15 Hz)	Clemson	Inertial	0.728	0.673	0.641	0.668
Two-stage CNN-LSTM [10] (our implementation, 8 sec @ 15 Hz)	Clemson	Inertial	0.783	0.680	0.697	0.683
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**

Test set results as reported in [8]. These models use time windows of 2 seconds, while single-stage models require 8 seconds due to their nature.

Specifically, we use the OREBA-DIS scenario with data for 100 participants (69 male, 31 female) and 4790 annotated intake gestures. The split suggested by the dataset authors [8] includes training, validation, and test sets of 61, 20, and 19 participants. For the inertial models, we use the processed[6] accelerometer and gyroscope data from both wrists at 64 Hz (8 seconds correspond to 512 frames). For the video models, we downsample the 140 × 140 pixel recordings from 24 fps to 2 fps (8 seconds correspond to 16 frames). For data augmentation, we use random mirroring of the wrist for inertial data and the same steps as [6] for video data, which includes spatial cropping to 128 × 128 pixels.

2) Clemson: The publicly available Clemson dataset [31] consists of 488 annotated eating sessions across 264 participants (127 male, 137 female), a total of 20644 intake gestures. Sensor data for accelerometer and gyroscope is available for the dominant hand at 15 Hz (8 seconds correspond to 120 frames). We apply the same preprocessing and data augmentation as for OREBA. 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 (see Supplementary Material S3). Note that because the Clemson dataset does not specify a dataset split, an alternative approach to test our models would have been k-fold cross-testing. However, we decided for a specific split approach because (1) there is no data scarcity in the Clemson dataset that would require k-fold cross-testing, and (2) applying k-fold cross-testing on the Clemson dataset would be prohibitively expensive. However, a shortcoming of this approach is that the results reported in Table II only reflect the test set which was selected by ourselves, not the original dataset authors.

D. Results

Results are listed in Table II, and extended results with detailed metrics are available in Supplementary Material S2.

1) Detecting Intake Gestures: The results for detecting only one generic intake event class are displayed in the center column of Table II. We can see that the single stage approach generally yields higher performance than the thresholding and two stage approaches: Relative improvements range between 2.0% (0.858→0.875) and 3.5% (0.781→0.808) over two-stage versions of our own architectures, and between 3.3% (0.783→0.808) and 10.4% (0.793→0.875) over our implementations of the SOTA.

For OREBA, we can additionally refer to previously published SOTA results based on 2 s windows. Relative improvements over these results for the inertial [10] and video [6] modalities equal 10.0% (0.778→0.855) and 2.6% (0.853→0.875), respectively.

For Clemson, we are not aware of any SOTA models other than the thresholding approach [4], [31]. It is not surprising that both the two-stage and single-stage approach outperform the thresholding approach by a large margin. Thresholding exclusively relies on one gyroscope channel, while the deep learning models build on a larger number of parameters. Consistent with the OREBA results, we find that the single-stage approach yields a relative improvement of 3.5% (0.781→0.808) over the two-stage models on the Clemson dataset. It is worth noting that the F_1 scores are generally lower for Clemson than for 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) Detecting Eating and Drinking Gestures: This task consists of localization and simultaneous classification of intake gestures as either eating or drinking. As there are no previously published results for this more fine-grained classification on either dataset, we rely on comparison between the separately trained single-stage and two-stage versions of our own models, as well as our implementations of the SOTA. 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.

Three main observations emerge. Firstly, the single-stage approach outperforms the two-stage approach to an even larger extent for this task: Relative improvements range from 1.9% (0.843→0.859) to 6.2% (0.783→0.832) over the two-stage version of our own architectures, and from 8.7% (0.765→0.832)

[6]Processing includes mirroring for data uniformity, removal of the gravity effect using Madgwick's filter [32] and standardization.

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	79.48	**0.48**	**10.53**	**0.15**	20.52	0.8341
Single-stage, extended prefix beam search	**80.58**	0.49	11.76	0.15	**19.42**	**0.8355**

to 17.7% (0.730→0.859) over our implementations of SOTA architectures. 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, an average decrease of 3.7% for OREBA and 7.3% for Clemson. Thirdly, there are generally 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.

Overall, the single-stage video models achieve the best results on OREBA. However, when focusing specifically on drinking detection, the two-stage video model achieves a better result. This may be due to the low number of drinking gestures in the test set, which causes F_1^D to randomly vary from $F_1^{E\wedge D}$ for multiple of the models in Table II.

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 to 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 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 s time window are plotted alongside the ground truth and predictions of the corresponding two-stage and single-stage models. Note that the output frequencies of the models differ, with 2 Hz for the video models and 8 Hz for the inertial models. We observe that the predictions by the two-stage models indeed mimic the ground truth, while the single-stage models produce probability spikes. Furthermore, these probability spikes line up temporally with the patterns that appear to be most distinct about the gestures for the human eye.

For a broader view of these characteristic differences between the single-stage and two-stage models, we use linear interpolation to aggregate the probabilities within all true positives in the validation set on a unitless timescale. The distributions displayed in Fig. 6 confirm that the two-stage models mimic the ground truth, while the probability spikes for single-stage models seem to be clustered in regions specific to the sensor modality. While the probability spikes for the video models tend to fall in the first half of the ground truth events, those for the inertial models appear mainly in the second half. This lends itself to the interpretation that video models target the frame in which ingestion takes place or the mouth is open (relatively early in the ground truth event), while inertial models leverage the characteristic downwards motion when finishing the intake gesture (relatively late).

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 predefined 2 s 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.

F. Difference Between Greedy Decoding and Extended Beam Search Decoding

Recall that greedy decoding only considers the maximum probability token at each time step, which is equal to extended prefix beam search decoding with a beam width of 1. As we increase the beam width, the algorithm considers more possible alignments and combines their probabilities if they lead to the same output sequence. In theory, this means that the results produced by the extended prefix beam search decoding with a higher beam width better reflect the network's intended output than greedy decoding, since they are computed in the same way as CTC loss works internally.

To analyze the effect of different beam widths on the F_1 score and determine a beam width to use in our experiments, we decode our trained networks with different beam widths on the validation set. As illustrated in Fig. 7, the effect of extended prefix beam search decoding is not very noticeable - a relative improvement of only 0.25% on average. In fact, there is no improvement for beam widths over 3, and hence we chose beam width 3 for decoding on the test set.

An explanation for these numbers may lie in the *few classes* (i.e., only one or two types of gestures to be detected) and the associated relatively *low uncertainty* exhibited by our scenario (i.e., limited variety of foods and environments). This is also indicated by the low rate of false positives of type 3 in Table III and the high prediction confidences in Fig. 5. It is well known that greedy decoding can work well as a heuristic in cases where most of the probability mass is allotted to a single alignment [27]. It is evident from Fig. 7 that higher beam widths mainly

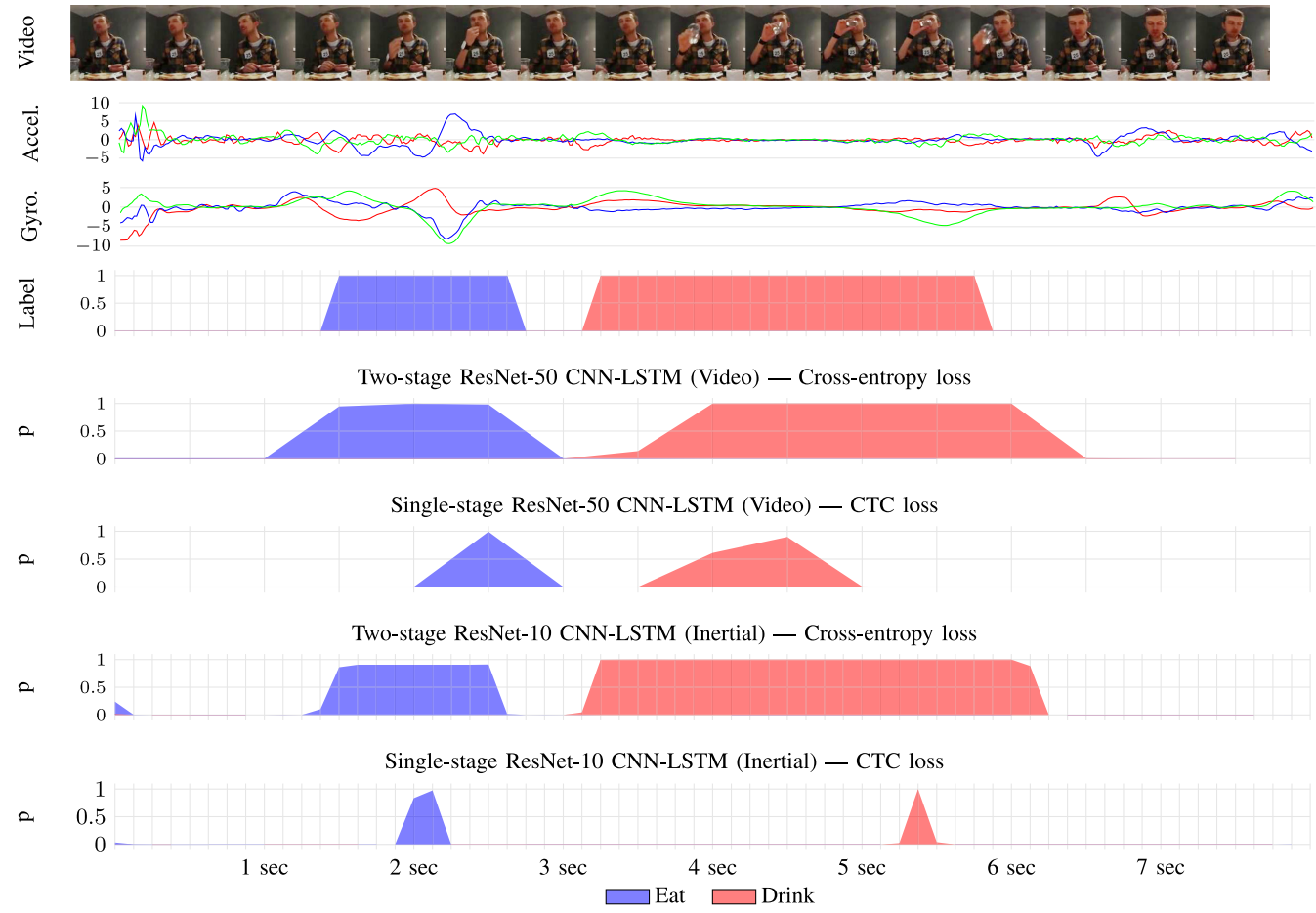

Fig. 5. Illustrating the effect of training with CTC loss or cross-entropy loss using input data, label, and model predictions for one 8 s example from the OREBA validation set.

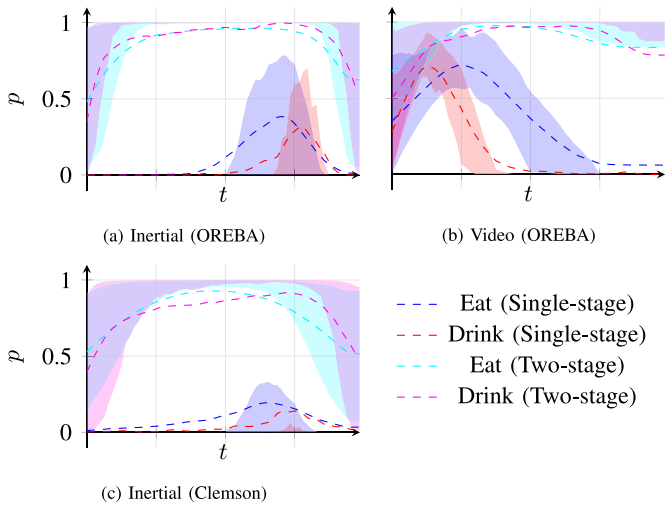

Fig. 6. Aggregating the predicted probabilities within all eating and drinking events in the validation sets of OREBA and Clemson. Probabilities are aligned in time and linearly interpolated, based on which we plot the mean and $[q_{25}, q_{75}]$ interval. The characteristic peaks for single-stage models trained on inertial data appear to be clustered in the second half of ground truth events, while they mainly fall in the first half for models trained on video data.

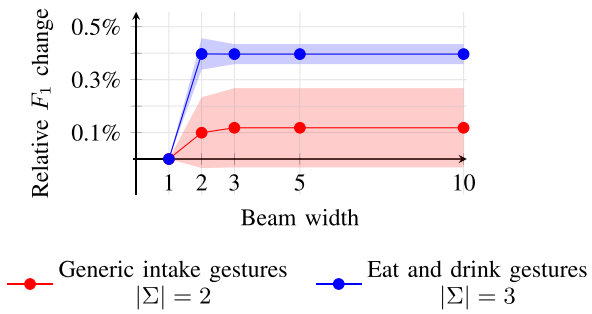

Fig. 7. Average relative F_1 change with standard deviation when choosing different beam widths for decoding our models on the validation set. The base scenario is a beam width of 1, which corresponds to *greedy decoding*. We observe that extended prefix beam search decoding mainly benefits the models for eating and drinking detection, and that there are no improvements for beam widths greater than 3.

benefited our task on eating and drinking gestures, which has one extra class and hence inherently carries more uncertainty. Following this line of thought, it seems likely that the extended prefix beam search algorithm could lead to higher benefits over greedy decoding for datasets with more diverse labels and scenarios.

V. Discussion

It is important to note that even though our implementations of the single-stage approach exhibit performance improvements compared to the two-stage approach, there are also several other differences between the two approaches that need to be considered in their application in research and practice.

First, the single-stage approach does not require detailed labels for the start and end timestamp of an intake gesture, but only a label for its apex. These simplified labels can assist in reducing the effort in labelling new datasets or applying the approach in contexts where there are constraints on the sampling rate of the ground truth label (e.g. time-lapse recordings in field settings).

Second, while the probabilities provided by the two-stage approach align closely with the entire duration of the intake gesture as provided by the ground truth label, the single-stage approach only yields individual spikes within the intake gesture (see Fig. 5). As such, the information provided by two-stage models is *richer* in the sense that they allow to estimate the duration of the gesture as well as the timing between gestures, which is not possible with the single-stage approach. For instance, if the spike in one gesture is towards the start of the ground truth event, and the spike in the subsequent gesture is towards the end, one would overestimate the gap between these gestures. In other words, the simplified labels of the single-stage approach come with the caveat of simplified information in its predictions.

Third, both approaches rely on specific yet different assumptions related to the duration of eating gestures. While the two-stage approach relies on a predefined gap between intake events (e.g., 2 seconds in [9], [6]), the single-stage approach requires a window that is sufficiently large to likely capture a sequence of at least two intake gestures (e.g., 8 seconds). The predefined gap of the two-stage approach creates the potential of inadvertently rejecting local probability maxima that are too close to each other. By contrast, the large window of the single-stage approach comes with the drawback of increased memory requirements which are also reflected in the choice of 2 fps for the video models.

VI. Conclusion

In this paper, we introduced a single-stage approach to detect 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, 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/drinking detection 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 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.

While we used the CNN-LSTM framework for our models, one could also consider alternative architectures. Importantly, the network must be able to cover the temporal context – this makes it difficult to directly combine CTC loss with convolution-only models such as SlowFast [33]. While CTC loss is traditionally combined with RNNs for this reason, Transformers have more recently emerged as another feasible choice [33]. Another topic to be explored in future research is the effect of choosing different window sizes on model training and performance.

This work also has several other implications for future research. We have shown a feasible way of localizing intake gestures while simultaneously classifying them as eating or drinking. Given larger video datasets with more different food types and associated labels, future research could explore more fine-grained classification of different foods and gestures. The necessity of large datasets has been pointed out [34] 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 detection. 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

This work was supported in part by Bill and Melinda Gates Foundation under Grant OPP1171389 and in part by Australian Government Research Training (RTP) Scholarship.

References

[1] G. Block, "A review of validations of dietary assessment methods," *Amer. J. Epidemiol.*, vol. 115, no. 4, pp. 492–505, 1982.
[2] S. W. Lichtman *et al.*, "Discrepancy between self-reported and actual caloric intake and exercise in obese subjects," *New Eng. J. Med.*, 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, pp. 1–28, 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," *Appl. Psychophysiol. 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 Anal. Process.*, 2017, pp. 411–418.

[6] P. V. Rouast and M. T. P. Adam, "Learning deep representations for video-based intake gesture detection," *IEEE J. Biomed. Health Informat.*, vol. 24, no. 6, pp. 1727–1737, Jun. 2020.
[7] J. Qiu, F. W. Lo, and B. Lo, "Assessing individual dietary intake in food sharing scenarios with a 360 camera and deep learning," in *Proc. Int. Conf. Wearable Implantable Body Sensor Netw.*, 2019, pp. 1–4.
[8] P. V. Rouast, H. Heydarian, M. T. P. Adam, and M. Rollo, "OREBA: A dataset for objectively recognizing eating behaviour and associated intake," *IEEE Access*, vol. 8, pp. 181 955–181 963, 2020.
[9] K. Kyritsis, C. Diou, and A. Delopoulos, "Modeling wrist micromovements to measure in-meal eating behavior from inertial sensor data," *IEEE J. Biomed. Health Informat.*, vol. 23, no. 6, pp. 2325–2334, Nov. 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," *IEEE Access*, vol. 8, pp. 164 936–164 949, 2020.
[11] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling," in *Proc. Eur. Conf. Comput. Vis.*, 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. 1168, pp. 1–25, 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. Biomed. Eng.*, vol. 56, no. 6, pp. 1663–1672, Jun. 2009.
[15] R. Zhang and O. Amft, "Monitoring chewing and eating in free-living using smart eyeglasses," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 1, pp. 23–32, Jan. 2018.
[16] R. Zhang and O. Amft, "Retrieval and timing performance of chewing-based eating event detection in wearable sensors," *Sensors*, vol. 20, no. 2, pp. 1–17, 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 J.*, vol. 12, no. 5, pp. 1340–1348, May 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. 209, pp. 1–16, 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. Biomed. Health Informat.*, to be published, doi:10.1109/JBHI.2020.2984907.
[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," *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 *Proc. Int. Conf. Document Anal. Recognit.*, 2007, pp. 1–5.
[24] Y. M. Assael, B. Shillingford, S. Whiteson, and N. De Freitas, "Lipnet: End-to-end sentence-level lipreading," 2016, *arXiv:1611.01599*.
[25] Q. Dai, J. Hou, P. Yang, X. Li, F. Wang, and X. Zhang, "Demo: The sound of silence: End-to-end sign language recognition using smartwatch," in *Proc. MobiCom*, 2017, pp. 462–464.
[26] Q. Zhang, D. Wang, R. Zhao, and Y. Yu, "MyoSign: Enabling end-to-end sign language recognition with wearables," in *Proc. Int. Conf. Intell. User Interfaces*, 2019, pp. 650–660.
[27] A. Hannun, "Sequence Modeling With CTC," *Distill*, 2017. Accessed: Jan. 21, 2021. [Online]. Available: https://distill.pub/2017/ctc
[28] *The TensorFlow Authors*, "TensorFlow API docs: Tf.nn.ctc_loss," 2020. Accessed: Jan. 6, 2021. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/nn/ctc_loss
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778.
[30] A. Hannun, A. L. Maas, D. Jurafsky, and A. Y. Ng, "First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns," 2014, *arXiv:1408.2873*.
[31] 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. Biomed. Health Informat.*, vol. 21, no. 3, pp. 599–606, May 2017.
[32] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," University of Bristol (U.K.), Tech. Rep. 25, 2010, pp. 1–32.
[33] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in *Proc. ICASSP*, 2019, pp. 7115–7119.
[34] Y. Shen, E. Muth, and A. Hoover, "The impact of quantity of training data on recognition of eating gestures," 2018, *arXiv:1812.04513*.