KU LEUVEN

FACULTY OF ENGINEERING TECHNOLOGY

CAMPUS GROUP T LEUVEN

Multi-Stage Temporal Convolutional Network for Fine-Grained Food Intake Detection:

Bite Detection with Free-Living Data

Jiayi WANG

Shuai ZHANG

Supervisor(s): Bart Vanrumste, Hans Hallez
Co-supervisor(s): Chunzhuo Wang

Master Thesis submitted to obtain the degree of Master of Science in Engineering Technology: Electronics Engineering

Academic Year 2021 - 2022

KU LEUVEN

Faculty of Engineering Technology Campus GROUP T Leuven — Master Thesis submitted to obtain the degree of Master of Science in Engineering Technology: Electronics Engineering
Academic Year 2021 - 2022

EMBARGO

This Master's Thesis is under embargo until 12-12-2030

Multi-Stage Temporal Convolutional Network for Fine-Grained Food Intake Detection:

Bite Detection with Free-Living Data

Zhang Shuai, Wang Jiayi

Master in Electronics Engineering, Faculty of Engineering Technology, Campus GROUP T Leuven, Andreas Vesaliusstraat 13, 3000 Leuven, Belgium

Supervisor(s): Bart Vanrumste, Hans Hallez

Electronics Engineering, Faculty of Engineering Technology, Campus GROUP T Leuven, Andreas Vesaliusstraat 13, 3000 Leuven, Belgium, <bart.vanrumste@kuleuven.be, hans.hallez@kuleuven.be>

Co-supervisor(s): Chunzhuo Wang

Electronics Engineering, Faculty of Engineering Technology, Campus GROUP T Leuven, Andreas Vesaliusstraat 13, 3000 Leuven, Belgium, <chunzhuo.wang@kuleuven.be>

ABSTRACT

Food intake monitoring techniques have been widely adopted in both civil and medical use, since the public have been more concerned about individual health situation than any time before. Several physical and mental disorders, including obesity, diabetes, fatty liver, sleep loss, etc. have been proven to be influenced by the food intake habits. Thus, the development of food intake monitoring techniques is of great importance. This paper proposes an approach to fine-grained detection with two-hand free-living data measured by inertial measurement units (IMU), including eating bites detection from in-meal data, meal-session interval segmentation and the corresponding bites detection from full-day data. Initially, the different models were studied and applied to detect bites of in meal-session data. Later, the segmentation of meal-session intervals was achieved by clustering and evaluating the density of bites within the intervals. Besides, Leave-one-subject-out (LOSO) validation of in-meal detections was performed on different models, obtaining segmental F1-scores for different Intersect over Unit (IoU) thresholds (0.10, 0.25, 0.50) of 91.3%, 91.2% and 87.7% on the proposed model Multi-Stage Temporal Convolutional Network (MS-TCN). Moreover, with LOSO validations on full-day datasets, the mean IoU score of 0.9 and mean segmental F1-scores of 88.1%, 83.9% and 63.8% were achieved on the proposed model. Finally, a publicly available dataset was used to further compare the performance of the proposed model MS-TCN and the Long Short-Term Memory (LSTM) model.

1 INTRODUCTION

Obesity has been publicly recognized as a global epidemic by the World Health Organization (Ulijaszek, 2003). As the living standards continue to rise in recent decades, weight gain and obesity have posed increasing threats to all countries globally. Besides, a number of other diseases, including type 2 diabetes mellitus (T2DM) (Khazrai et al., 2014) and fatty liver diseases (Trovato et al., 2015), are raising concerns of not only adults, but also children and teenagers. Research has shown that one's dietary habits is a major factor for the rising incidence of diabetes among many countries (Organization, 2016). The dietary patterns have also been proven to affect mental well-being of adolescents (Oddy et al., 2009) and account for sleep loss for adults (Crispim et al., 2007). There is no doubt that one's food intake behavior has significant impacts in all aspects of life. Thus, it is of great importance for constant food intake detection for both health and societal purposes.

Currently, the civil use of food intake monitoring involves extensive self-report activities, which could not only be troublesome and inefficient, but also causing unpredictable errors due to the proximity of human judgment (Schoeller, 1995). Such errors in self-report could cause serious consequences among specific groups, such as obese patients (Lansky & Brownell, 1982). Therefore, the implementation of automated food intake detection is a significant task.

In this paper, a comprehensive framework for automated food intake detection was proposed using collected inertial data. Data were collected with a new experiment design, acquired from 21 subjects. It was followed by data segmentation, preprocessing and annotation. Furthermore, a new approach using Multi-Stage Temporal Convolutional Network (MS-TCN) model was introduced and compared to other existing models. Then, a new evaluation scheme using segmental F1 scores was proposed. By determining the meal intervals from the full-day dataset, we achieved the meal session allocation. In addition, the bites within the corresponding intervals were detected to finally achieve fine-grain bite detection in free-living conditions. Besides, results of in-meal detection from the proposed MS-TCN model were compared to the publicly available dataset FIC to draw conclusions.

This paper is organized as follows. Section 2 reviews relevant work concerning various food intake detection techniques and the remaining research gaps. Section 3 briefly introduces the fundamental techniques further adopted in this study. Section 4 explains the necessary methods and approaches to achieve the bite detection. Section 5 demonstrates experiments on datasets and corresponding results. Section 6 presents discussions on improvements and limitations of the proposed approach. Section 7 draws a brief conclusion to the paper.

2 LITERATURE REVIEW

2.1 Related Work

Sensors have always been an important tool in action detection studies. To achieve human activity recognition, various sensors have been used in previous studies. In this section, previous works will be introduced based on the type of the sensors used in the study.

2.1.1 Acoustic sensor

A wearable non-invasive acoustic sensor can be used to collect sound data. The swallowing sound is detected and classified based on mel-scale Fourier spectrum features and support vector machines (SVM). Then, principal component analysis and a smoothing algorithm are used to improve the detection accuracy (Makeyev et al., 2012). Hidden Markov models are also used for the recognition of single chew or swallowing events (Päßler et al., 2012). The work by Lopez-Meyer et al. (2010) utilized SVM to detect the food intake process. It was observed that compared to accuracy (80%) obtained by using swallowing as the sole predictor, higher accuracy can be achieved when both chews and swallows are used as predictors. The work presented by Fukuike et al. (2014) and Pasler and Fischer (2014) also conducted swallowing detection in free-living conditions.

2.1.2 Accelerometer

Lin et al. (2012) used the data collected from a triaxial digital accelerometer with one electrocardiogram (ECG) to build a radial basis function network (RBFN) and a generalized regression neural network (GRNN) as energy expenditure regression (EER) models and compare their performance towards human activity classification. The work presented by Dongwoo and Kim (2007) use multi-site triaxial accelerometers mounted on human wrists, ankles and body drunks to calculate the integral value and improve the EER performance. Similarly, different types of accelerometers are used to achieve EER in other papers (Yang & Hsu, 2009)(Kang et al., 2010).

2.1.3 Piezoelectric Strain Sensor

Farooq and Sazonov (2016b) used a piezoelectric strain sensor to collect data related to the jaw activity from piezo-

electric strain sensor and accelerator, and to train SVM classifiers for food intake and activity detection. The results from SVM were also combined using a decision tree (two-stage classification) to determine the final class. Similar jaw activity detection models based on piezoelectric strain sensor are presented in other studies (Farooq & Sazonov, 2016a)(Farooq & Sazonov, 2017).

2.1.4 Mandometer

Mandometer monitoring plate weight has also shown promising results in food intake detection. In the study of Papapanagiotou et al. (2015), they proposed a novel algorithm called parametric Probabilistic Context-Free Grammar (PCFG) for automatically constructing a subject's cumulative food intake (CFI) curve to detect food intake cycle and calculate intake weight using only the mandometer weight measurements. Similar studies also used mandometer to generate CFI curve to detect the food intake cycle and consumption by algorithms (Papapanagiotou et al., 2019)(Mattfeld et al., 2017).

2.1.5 Video Camera

The works of M. M. Anthimopoulos et al. (2014), W. Zhang et al. (2015) and M. Anthimopoulos et al. (2015) used a computer vision-based model to recognize food type and estimate a meal's carbohydrate content for diabetic patients. Lea et al. (2017) used a temporal convolution network to achieve action detection and segmentation. It also proves that temporal convolutional network (TCN) can capture longer memory than LSTM-based Recurrent Neural Networks even in challenging fine-grained datasets.

2.1.6 IMU Sensor

S. Zhang et al. (2018) integrated IMU, proximity sensor, and ambient light sensor to form the neck-worn sensor. It can be used to detect chewing counts, number of feeding gestures, postures, and activity intensity. The results of monitoring can be displayed in real time on smartphones. Kyritsis et al. (2019) proposed a micromovement architecture for eating behavior modeling. In this study, the wrist actions related to food intake were classified as five micromovements: pick up, upwards, downwards, mouth, no movement. The collected IMU data were input into CNN to obtain the micromovement probability distributions. Then, the distributions were input to the LSTM to get the classification result. Due to the complexity of the annotation for micromovements, the author recently proposed a complete eating behavior detection framework in the study based on a 6-axis IMU unit (Kyritsis et al., 2021). The framework included fine-grained detection of each bite for in-meal session dataset and a segmentation for meal-session intervals for free living dataset. A model implementing CNN and LSTM were applied to achieve aforementioned fine-grained detection and segmentation. Besides, an algorithm based on Gaussian filtering was proposed for segmentation.

2.2 Research Gaps

As described above, although many advanced algorithms and models have been proposed, there are still many research gaps that need to be solved.

2.2.1 Simultaneous Detection on Both Hands

In the work of Kyritsis et al. (2021), data from only one hand was collected. Data from left-headed subjects were mapped to the corresponding right-handed coordinate systems. In actual meal sessions, alternating use of both hands often occurs. In another group's work, two 6-axis IMU sensors were used to collect data from both hands. Then the data of 12 axes were simultaneously fed into the CNN as 12 input channels (Rouast et al., 2020). The final F1 result of 85.2% was obtained. However, since the information of the 12 channels is input into the model simultaneously, the information of both hands was mixed together and thus interfered. This paper proposes a method using offset-appending to input the information of both hands into the model without interference.

2.2.2 Segmental Evaluation Metrics

In both works mentioned in section 2.2.1, only the detection performance was evaluated while the segmentation performance was not (Rouast et al., 2020). In this study, the $F1@k$ score was used as the evaluation metrics to assess the performance of the models under certain IoU-threshold value k, as further explained in section 4.9. Both detection and segmentation capabilities were assessed.

By evaluating the segmentation performance, a more incisive assessment could be drawn for the model. The model with better segmentation performance can usually detect the boundaries of bites more precisely, and thus achieve fine-grain detection by determining each bite duration. Researches have shown that a number of diseases may have an impact on food intake speed, including Parkinson's (Norberg et al., 1987). Therefore, the detection of bite duration could play an important role in the diagnosis of such diseases.

2.2.3 TCN Modeling

Almost all related works currently use sliding window to implement sequence modeling, and use LSTM to capture the temporal dependencies between gestures. Bai et al. (2018) have shown that TCN can achieve better performance than LSTM even on the best performing domains of recurrent network, and proved that the theoretical "infinite memory" of recurrent neural network (RNN) is largely absent in practice. The work of Lea et al. (2017) implemented human activity recognition and segmentation tests based on video signals using TCN modeling. Still, research gap remains in TCN models based on IMU signals.

2.2.4 Free-living Detection

A segmentation approach based on Gaussian filtering was proposed by Kyritsis et al. (2021). Although segmentation of meal intervals in long-period data was achieved in this approach, fine-grain bite detection within meal intervals was not achieved. Besides, with Gasssian filter and thresholding operation for segmentation, even if the side-lobe detector was added, the boundary was damaged.

Another segmentation approach using top-down architecture was propose by Sharma and Hoover (2022). While the approach proposed by Kyritsis applied clustering after detecting each bite to form meal intervals, this approach achieved meal interval detection directly with two thresholds, as shown in Figure 2.1. The first threshold T_s is a high value for detecting the start of the meal interval. By choosing a high threshold value, false positives could be eliminated, so as to reduce the impact of noise. The second T_e is a low value for detecting the end of the meal interval. As per Sharma, eating behaviors tend to be more vigorous at the start of a meal and fade with satiety (Sharma et al., 2020)(Dong, 2012). Thus, by choosing a low threshold value, the weaker probability at the end of the meal could be detected so as to increase boundary precision. However, several research gaps remained in this approach. First, the top-down architecture only achieved meal interval detection, but did not further detect the bites within meal intervals. Second, although the detection of false positives could be avoided by choosing a high T_s value, some true positive bite behavior with lower probability score would be ignored at the start of the meal intervals. Third, since the database Clemson All-Day Dataset (CAD) used in this study was collected on single-hand 6-axis IMU sensors, the fact that subjects sometimes ate meals with alternating hands was not taken into account (Sharma, 2020).

In this paper, a new approach using bottom-top architecture was proposed for free-living detection. The bites were first detected respectively. Then, meal intervals were clustered by morphological closing, thus without damaging the boundaries of the intervals. The left and the right boundaries of the meal interval were determined by the start of the first bite and the end of the last bite within the corresponding meal interval, respectively. Finally, false-positive detection caused by noise was eliminated by density filtering. Besides, this new approach achieved detection of bites in meal sessions using alternating hands and fine-grained bite detection after having determined the boundaries of meal intervals.

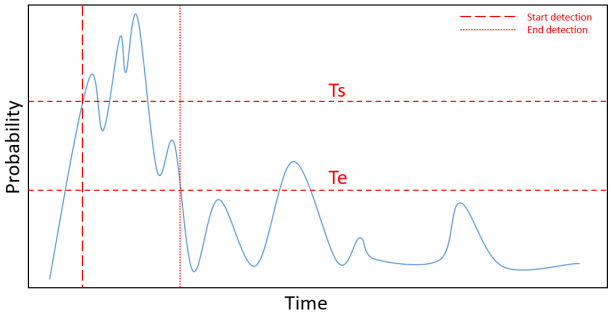

Figure 2.1: Meal Interval Detection with Two Thresholds T_s and T_e

3 RELATED TECHNIQUES

In this section, several techniques used in this study will be briefly introduced to make the model structure easier to understand.

3.1 CNN

Convolutional Neural Network (CNN) is a model commonly used for spatial feature extraction (Zhao & Du, 2016). The basic composition unit of CNN is the filter. For each filter, there are three hyper-parameters: filter size, filter stride and activation function. Filter size determines the receptive field. Filter stride determines the step size along the input sequence. Activation function defines the non-linear processing scheme. Figure 3.1 demonstrates a 1-D convolution operation. After a convolution operation with three channels, each filter will produce a new value to represent the operation result for the corresponding interval. Then the sliding window is generated along the stride to obtain the subsequent points. Finally, the endpoint is supplemented by zero padding to obtain a new sequence of five points. Each point contains the spatial feature information of the previous sequence within the respective interval. Because of the overlap of the sliding windows controlled by the stride, CNN can search through the complete sequence.

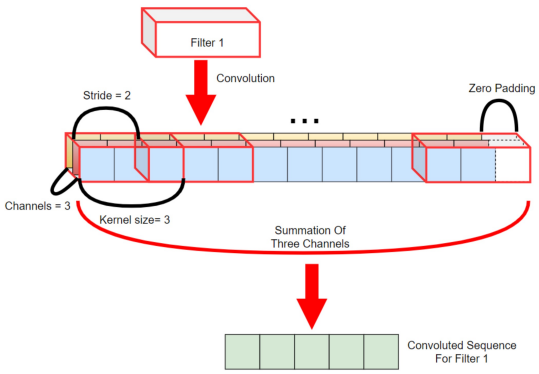

Figure 3.1: Convolution Operation on 1-D Sequence

3.2 Max Pooling

Pooling is a sequence processing technique used to reduce parameters and computational costs. Pooling usually includes average pooling and max pooling. Figure 3.2 illustrates a max pooling process with a pooling size of 2, which means the maximum value of each two adjacent points is retained and used to represent the characteristic strength for the corresponding position. By keeping the maximum value among a certain pooling size, max pooling can reduce the sequence length without information loss, since the resolution of the information would be lost anyhow, so as to reduce the computational costs in the following layers of the network.

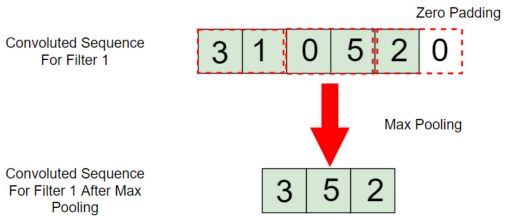

Figure 3.2: Max Pooling

3.3 Dilated Convolution

Dilated convolution is a variant of CNN (Yu & Koltun, 2015). With the same filter size and computational costs, dilated convolution provides a wider range of receptive fields, detecting longer history in the temporal sequence. As shown in Figure 3.3, with the same Kernel size and stride, the receptive field with a dilation factor of 2 is twice as large as the field without dilation. In deep networks, dilation factor usually increases exponentially with the number of layers, thus further expanding the receptive field.

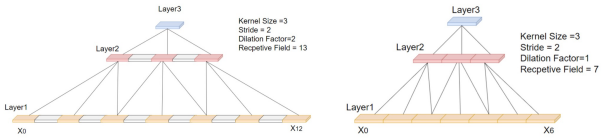

Figure 3.3: Dilated Convolution

3.4 Recurrent Neural Network (RNN)

Recurrent neural network (RNN) is a neural network model used for temporal signal processing. Compared with normal neural networks, RNN can capture the temporal context of sequence signals. Thus, even with the same input signal, differences in signal sequences will result in different outputs. Figure 3.4 illustrates the structure of the RNN model, where $x_t(t = 1,2,3)$ is the current input signal at time t; $h_t(t = 1,2,3)$ is the current hidden state at time t derived from the previous hidden state h_{t-1} and the previous input x_{t-1}); $y_t(t = 1,2,3)$ is the current output signal at time t; w_i, w_o and w_h are the parameters of the neural network derived through training. Thus, the output for each moment is determined not only by the current input signal but also by the hidden state which has considered all previous inputs and sequences. The main problem of

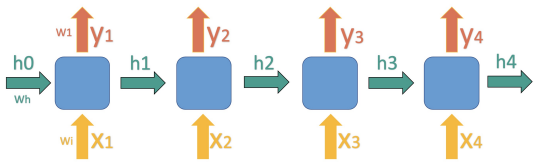

Figure 3.4: RNN Architecture

RNN is that it cannot achieve long-distance memory. The hidden state is forced to be updated at each moment. After a few stages of calculation in the nodes of the neural network, the previous features of a longer period will have been covered. In addition, the problems of both vanishing gradient and gradient explosion exist during the training of RNN. Therefore, a special RNN model named Long Short-Term Memory was used, as explained in section 3.6.

3.5 Bi-directional RNN

Compared to normal RNN architecture, bi-directional RNN (Bi-RNN) has an output that is determined by hidden states in both forward and backward directions as shown in Figure 3.5, thus achieving the non-causal effect.

For the current moment t, being causal means the current output depends on the input sequence $[t-n,t]$, while being non-causal means the current output depends on the input sequence $[t-n,t+m]$, which will cause information leakage accordingly. Figure 3.6 demonstrates the architectures of both casual and noncausal models.

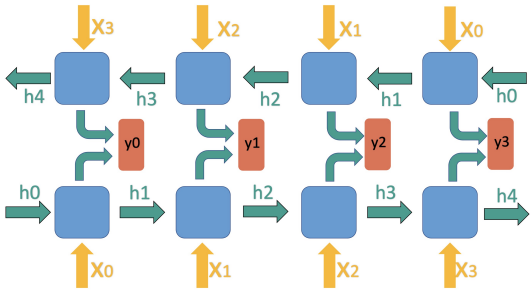

Figure 3.5: Bi-RNN Workflow

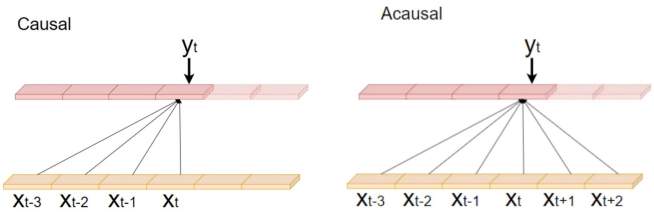

Figure 3.6: Architecture of causal and noncausal mode

3.6 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a special type of RNN that solves vanishing gradient and gradient explosion problems in the long sequence training process (Hochreiter & Schmidhuber, 1997). As shown in Figure 3.7, compared with traditional RNN which has only one transfer state (hidden state), LSTM has two transfer states (cell state C_t and hidden state h_t) and three more inputs Z_f, Z_o and Z_i, controlling respectively three gates: input gate (controlling input), forget gate (controlling previous cell memory removal) and output gate (controlling output). Due to the existence of the three gates, the C value will not be forced to change with the time step (when the input gate is closed), which achieves selective memory, such as only remembering the key information and retaining it for a long time without being destroyed, unlike the information added up without selection in a normal RNN model. Similar to RNN, LSTM also has its causal and noncausal mode. The noncausal LSTM is implemented by Bi-directional Long Short-Term Momory (Bi-LSTM), which has similar architecture as Bi-RNN.

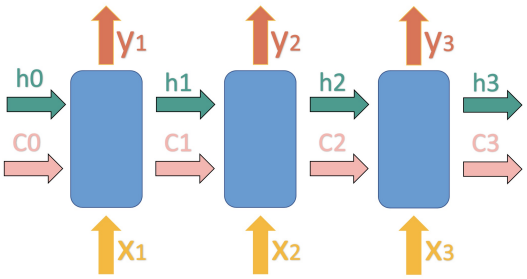

Figure 3.7: LSTM Architecture

The workflow of LSTM is as Equations 3.1 and Figure 3.8:

$$\begin{aligned} C_t &= Z_f * C_t - 1 + Z_i * Z \\ h_t &= Z_0 * \tanh(C_t) \\ y_t &= \sigma(W' * h_t) \end{aligned} \tag{3.1}$$

i) The input x_t of the LSTM and the hidden state h_{t-1} of the previous moment are computed to obtain four inputs of three gates and the processed inputs;

ii) The new cell memory C_t is obtained through the forget gate and input gate and the processed input z;

iii) The hidden state h_t and the output y_t are calculated by C_t and output gate.

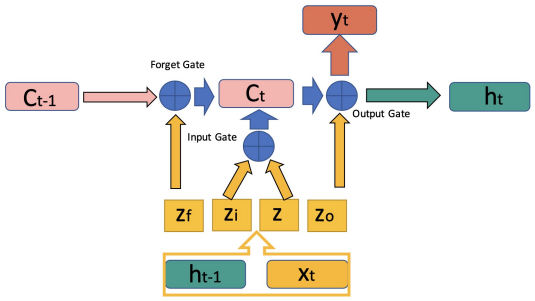

Figure 3.8: LSTM Workflow

3.7 Attention

In a traditional attention model, a query and a key are generated for each input (Query-Key-Value structure). The corresponding output value is obtained by computing the weighted average of multiple inputs, while the weights are determined by similarity computation between the queries and keys (Vaswani et al., 2017). However, the attention model presented here is different from the traditional attention model based on the Query-Key-Value structure, which is a simplified version composed of learnable functions (Raffel & Ellis, 2016).

Attention can provide direct and selectable dependence between the state of the model at different points in time. As shown in Figure 3.9, for each input (either an embedding vector, or the hidden state of RNN), a learnable function a that scores the inputs at each moment is executed (the scoring can be interpreted as the similarity of the inputs between each point). Then the results of these scores were applied to a Softmax calculation to obtain the weighted α of each input. Finally the weighted mean of the input sequence is obtained by weighting the weights to sum up.

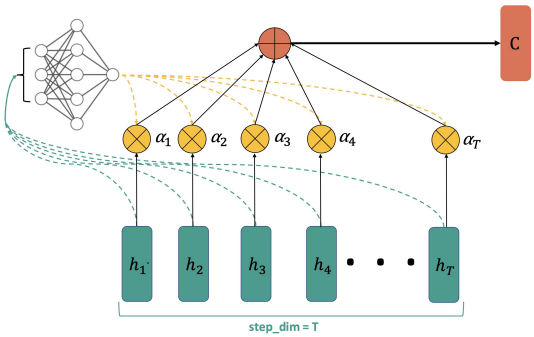

Figure 3.9: Attention Model Architecture

3.8 Residual Connection

The residual connection is a model optimization method (He et al., 2016). It is commonly used in the neural network with deep layers to solve the problem of vanishing gradient. The structure is shown in Figure 3.10:

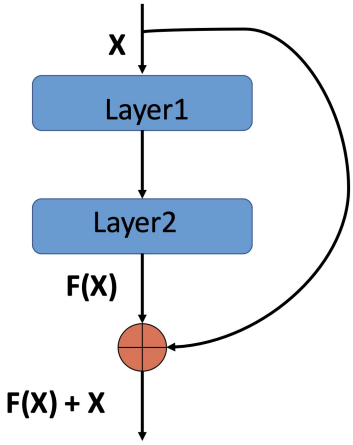

Figure 3.10: Deep Network with Residual Connection

The output has an additional value of input x after the structure of the residual connection. The advantage of this method is that the back propagation of the output has an additional constant term, so that the problem of fast convergence of the gradient to zero can be avoided when multiplied by the chain rule.

3.9 Morphological Closing

Morphological Closing is a 2-D image processing operation commonly used to fill up cracks and small holes (Vincent, 1994). For simplicity, only 1-D operation will be introduced here. Closing is made up of two steps:

i) Dilation: the sequence is traversed by a kernel with size of 3. In the process of sliding the window, the middle element is replaced by the value of the largest element in the window, and the value of the step represents the number of times that the sequence is traversed, with a default of 1, as shown in Figure 3.11.

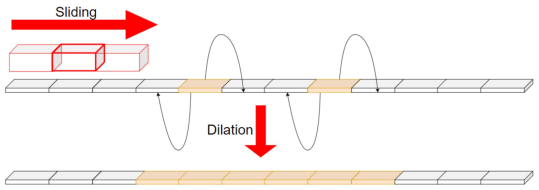

Figure 3.11: Dilation Workflow

ii) Erosion: the sequence that has been conducted with dilation operations of step times is used as the output. The kernel of the same size is used for the traversal operation. Each time the element in the middle position is replaced by the value of the smallest element in the window. For a step size of N. the corresponding processing flow chart is shown as Figure 3.12:

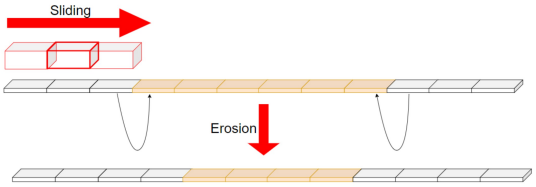

Figure 3.12: Erosion Workflow

In short, the dilation operation is used to fill the small holes, while the erosion operation is used to eliminate the excess tail generated by the dilation operation. Figure 3.13 demonstrates the workflow of morphological closing with a step size of N.

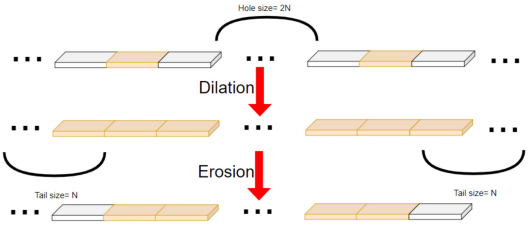

Figure 3.13: Morphological Closing Workflow

3.10 Offset Appending

Offset appending is a processing scheme proposed in this study to integrate data from both hands into one data stream. The principle of offset appending is shown in Figure 3.14. For simultaneous measurements with multiple sensors of the same type, sometimes it is desired that the measurements from each sensor should be input separately. Since measurements were made on both wrists with two IMU wristbands in this study as explained in section 4.2, in-meal data measured on the left hand were ap-

pended to the right-hand data by adding an offset, which equals the last timestamp of the right-hand data.

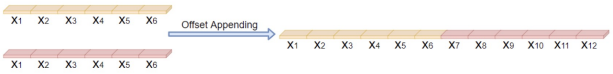

Figure 3.14: Offset Appending Scheme

4 METHODOLOGY

4.1 Sensor System

In this study, a commercially available IMU development kit ("ConsensysPRO Software", n.d.) was used, as shown in Figure 4.1. The Shimmer3 IMU consists of 9 Degrees of Freedoms (DoF) inertial sensing via 3-axis accelerometer, gyroscope and magnetometer, among which only the former two were deployed in this study. The built-in accelerometer measures accelerations on three axes in selectable ranges, among which $[-2g, 2g]$ was used in this study. The gyroscope on the MPU-9150 chip from Invensense is used to measure angular velocities in three dimensions. In this study, the sampling frequency of the IMU was set to 64Hz. The data collected by the IMU are stored in a built-in micro-SD card, which can be retrieved using desktop application Consensys ("ConsensysPRO Software", n.d.), and was preprocessed using MATLAB ("MATLAB - MathWorks", n.d.) before being used in model training.

Figure 4.1: Shimmer3 Sensor Unit

4.2 Experiment Design and Data Collection

This study has obtained approval (G-2021-4025) from the Social and Societal Ethics Committee (SMEC) of KU Leuven in the privacy and ethical review. An informed consent form was distributed to each subject prior to the participation of the study. The experiment was performed by each subject twice, either during two meals on the same day, or lunchtime on two different days. In addition, each subject is requested to perform a full-day experiment for a period of at least six hours. The full-day experiment may include two meal sessions, if the subject decided to perform the experiments in one day, or only includes one meal session, if the subject decided to perform the experiments in two different days.

For each experiment, regardless of the time duration of the participation, the subject was requested to put on two IMU wristbands over his or her wrists prior to the start of the experiment. A video camera was placed on the table in front of or beside the subject during each meal session. The camera aimed at the subject and the plate on which the food was placed. Only the food, the cutlery used and the upper body (higher than the table) of the subject was recorded. The video was only used to assist in annotating ground truth for the evaluation of validating automated detection of bites. At the start of each video recording, the current Unix timestamp was shown at the camera to provide proof of synchronization. Then the subject was requested to rotate his or her arm around the elbow on each arm, respectively, and finally rotate both arms at the same time. Such movements were used to provide proof for the time synchronization, together with the timestamp shown in the video. The demonstration of these movements are shown in Figure 4.2, in which a subject performed the requested movements in front of the camera, and the data were aligned along the video to fulfill synchronization of both. The corresponding waveform of the movements is shown in Figure 4.3.

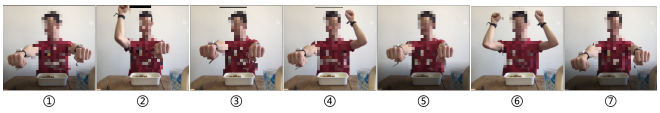

Figure 4.2: Wrist Movements for Synchronization

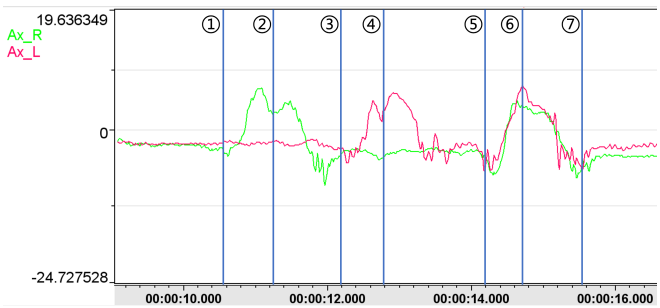

Figure 4.3: The Waveform Pattern for Synchronization, where Ax_L and Ax_R represent the x-axis acceleration measured on the left and right hand, respectively

A total of 21 subjects (14 male and 7 female, age ranging 19-27 years old) were recruited. This includes 17 subjects who use forks, knives or spoons as predominant tools for food intake and 4 subjects who use chopsticks and spoons as predominant tools. In total, 48 meal sessions (total length of 13.86 hours, average duration of each meal

1039.63 seconds) and 27 full-day sessions were monitored (3 subjects performed 2 full-day experiments with 2 meal sessions per day, 14 subjects performed 1 full-day experiment with 2 meal sessions per day, 4 subjects performed 1 full-day experiment with 1 meal session per day and 1 meal-only experiment) from December 2021 to March 2022. During each in-meal food intake session, the food was placed on a plate or in a bowl. The food used for the study (2 two-course meals with soup and main dish; 28 meals with a single dish, e.g. a salad, a portion of noodles or a portion of vegetable stew; 18 meals with staple and side, e.g. meatballs sided by baked potatoes, or stir-fried wok sided by rice) and the location of the meal (11 meals at university canteen, 37 at home or other private environment) were not restricted. The subject was then instructed to eat naturally, as he or she normally would, taking as much time as he or she preferred and pausing whenever he or she would like to. The subject was also given a glass of water, which he or she could choose to drink or not. The subject could use either hand for all activities, regardless of dominant or non-dominant hand. The goal of the in-meal food intake sessions was to train the model for automated bite detection.

In-Meal Sessions	**Values**
Number	48
Total Duration	13.86 hours
Average Duration	1039.63 seconds
Total Number of Bites	2599
Average Bites in Each Meal	54.14
Total Duration of Bites	6091.7 seconds
Average Duration of Each Bite	2.34 seconds
Average Duration of Bites in Each Meal	126.91 seconds
Ratio Bite/Meal (Bite Density)	0.0629

Table 4.1: In-Meal Dataset Statistics

For each full-day experiment, the subject was requested to put on the wristbands before the first meal session and to keep the wristbands on until the wristbands had recorded at least 6 hours' data. The subject performed daily tasks as what he or she would normally do in a free, unrestricted living environment. Such testing environments included schools, offices, shops, restaurants and other social settings, as well as homes. The subject was asked to remove the wristbands when the IMU had recorded at least 6 hours of data. The goal of full-day data collection was to enable automated meal-session detection, given a longer set of data for about 6 hours of in-the-wild unrestricted behaviors.

4.3 Signal Processing Algorithms

After the data collection of the subject for one full-day or in-meal session, data stored in the IMU were exported and preprocessed, before being used for automated detection. Each set of raw data obtained includes 12 columns of temporal data along with the timestamp. In this study, 6 degrees of freedom (DoF) inertial data were used, including high-precision accelerometer and gyroscope measurement in 3 streams (x, y and z streams), denoted as a_x, a_y, a_z, g_x, g_y, g_z, respectively.

4.3.1 Segmentation

For each meal session, the data were trimmed to a certain length in time, corresponding to the duration of the meal session. The duration of the meal session t_m was determined by the length of video recording, rather than the actual eating time, which was desired for annotation later. The starting time of the data segment was initially determined by reading the timestamp in the video recording. The length of the recording segment M was determined by $M = t_m * f_s$, where f_s is the sampling frequency of the IMU.

For each moment in time i within the recording segment, the row vector $A(i) = [a_x(i), a_y(i), a_z(i), g_x(i), g_y(i), g_z(i)]$ contains real-time line acceleration and angular velocity on 3 axes, whereas a_x, a_y, a_z are the accelerations on 3 axes and g_x, g_y, g_z are the angular velocities on 3 axes, respectively. A complete data segment can be represented as $R = [A(1), A(2), ..., A(M)]^T$.

4.3.2 Data Mirroring

In this study, the subjects were expected to use either hand to perform eating tasks with cutlery. Both left-handed and right-handed subjects were recruited. The subject was not restricted to eating with one specific designated hand, but was free to use either or both hands as he or she usually did. Since recordings on both hands were used for automated detection as independent data, it was desired that data on both hands share the common relative coordinate system (Kyritsis et al., 2021). The right hand of the subject was selected as the reference because the majority of the subjects were right-handed (16 right-handed, 5 left-handed). For each set of left-hand data, the a_x, g_y and g_z streams were inverted. This was achieved by inverting the x-axis of the IMU on the left hand. When observing the orientation of the axes on both wristbands, it can be revealed that the y- and z-axis do not need to be adjusted, while the x-axis should be adjusted so that when a wrist movement towards or away from the torso of the sub-

ject, the accelerometers of the IMUs on both wrists should yield the same measurement on x-axis. For instance, a movement towards the torso would yield positive measurements for the IMU on the right wrist, but negative measurements yielded on the left wrist. Thus the x-axis acceleration stream ax on the left wrist should be inverted. The angular velocity on y- and z-axes on the left wrist should be inverted for the same reason.

4.4 Annotation

4.4.1 Time Sync

When the meal-session data was segmented, data were annotated with the time of each bite. Though the timestamp shown to the video camera to provide a rough time estimation, due to the unavoidable time gap from the start time of the video to a certain second and the delay in the display of website refresh, there was bound to be a minor time difference between the exact start of the video and the first timestamp of the segmented data. Therefore, a time-sync step was needed. The movements which the subjects performed before starting to eat in a meal session (the rotation of arms) were used to provide a reference to the time sync. This sequence of movement was designed to have a fixed pattern in the waveform of acceleration stream, as shown in Figure 4.3 (the sequence corresponds to Figure 4.2). By aligning the movement observed in the camera and the waveform, accurate time synchronization can be achieved, with a visual judgment error below 100 milliseconds.

4.4.2 Annotation

There are a number of bites during each meal session. A bite is defined by its start and end. The start of a bite was defined as the moment the subject raised the cutlery from the plate or the bowl. The end of a bite was defined as the last moment during which the subject put down the cutlery for the first time after taking the cutlery out of the mouth. The number of bites in a meal session varies from 19 to 110 on different subjects, with an average and standard deviation of 54.21 and 19.27. Each bite was labeled with its start and end. A total of 2599 bites were annotated, with 1370 (52.71%) on the left hand and 1229 (47.29%) on the right hand. The distribution of bites on each hand in each meal session is shown as Figure 4.4.

The annotations were made using ELAN ("Download ELAN 6.3.0", 2022). The food pick-up process might not be annotated as part of a bite, since the behavior of picking up food varied vastly due to the type of food consumed,

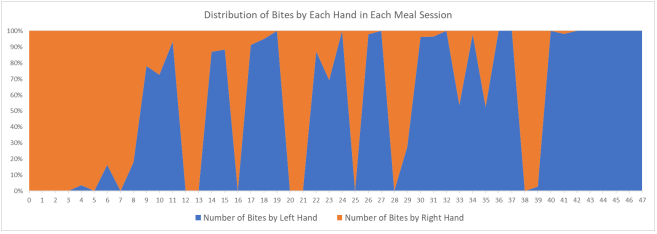

Figure 4.4: Distribution of Bites by Hand in Each Meal Session, where the x-axis represents the ID of each meal session, and the y-axis represents the distribution of bites for each meal session on both hands

the eating habit of the participant, as well as other factors. Figure 4.5 shows a demonstration of a bite annotation. For

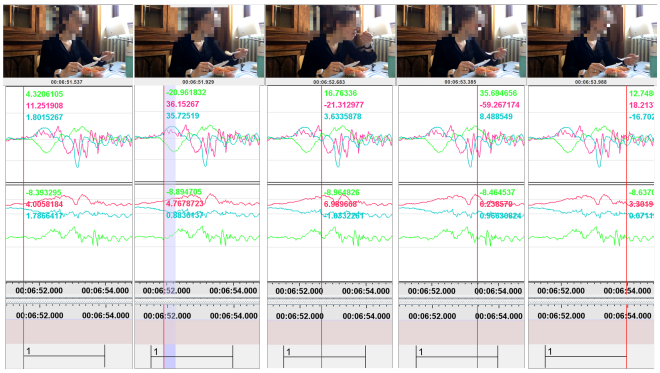

Figure 4.5: Waveform and Video of Bite Annotation, where the upper waveforms represent the gyroscope measurements and the lower waveforms represent the accelerometer measurements; the red, green and blue line represent measurements from the x-, y- and z-axis, respectively; same applies to Figures 6.1, 6.2 and 6.3

each bite carried out with a fork, a knife or a spoon, an annotation with number 1 was cast. For each bite with a chopsticks, an annotation with number 3 was cast. In the rare cases of bites served by fingers, an annotation with number 4 was cast. Different annotations were marked out for further categorization. Drinking behavior was also annotated with number 2, from the moment of tilting the glass to the moment that the glass was put back vertical. The categories of annotations are as Table 4.2:

Type of bite	Annotation
Bite by fork, knife or spoon	1
Bite by chopsticks	3
Bite by fingers	4
Drinking behavior	2

Table 4.2: Annotations for Different Bites

4.5 CNN-LSTM

4.5.1 Preprocessing

As the initial IMU data sequences were too long to be input into the model, the 6 DoF IMU data were appended (left-

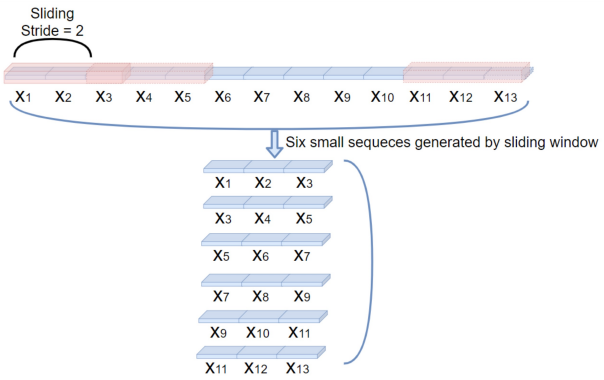

Figure 4.6: CNN-LSTM Preprocessing

hand data were appended to the end of the right-hand data of the same session), and then segmented and marked with windows. Figure 4.6 demonstrates a simplified process with a sliding window size of 3 and a sliding stride of 2, in which the original sequence with a length of 13 would generate 6 segments each with a length of 3. (The parameters that were finally decided on were a window size of 300 samples and a stride of 20 samples, which will be explained in subsequent chapters.) Based on the continuity of the eating behavior, the middle point mark was selected to represent each segment, which ensures the halving principle. As shown in Figure 4.7, if the middle point of a segment is marked as True, the whole segment will be marked as True, and vice versa.

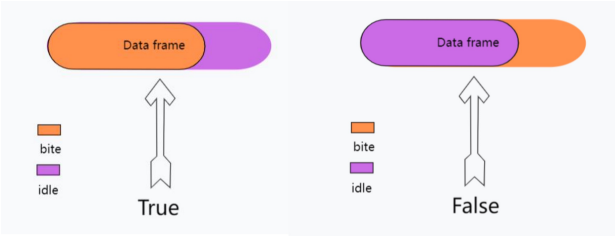

Figure 4.7: Middle Point Labeling

4.5.2 Architecture

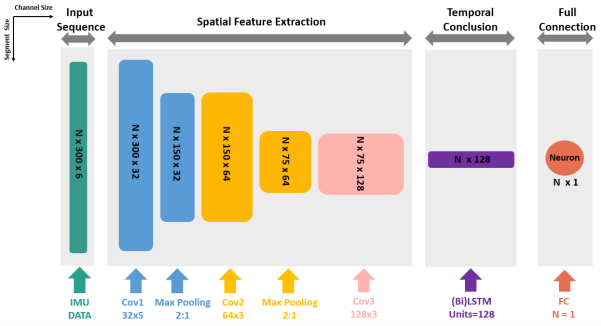

Figure 4.8: CNN-LSTM Architecture

As shown in Figure 4.8, there are four modules in the model. The horizontal direction represents the channel size. The vertical direction represents the segment size. The first module is the preprocessed $[N*300*6]$ IMU data segments, where N is the batch size, 300 is the segment length, and 6 is the number of axes of IMU data. The second module contains three convolutional layers and two max pooling layers. The convolutional layers (Relu selected as activation function) are used to implement spatial features extraction (e.g. the first convolutional layer $cov1$ contains 32 filters, each filter with a kernel size of 5. Thus, the input data are processed into a new sequence of $[N*300*32]$). The max pooling layers are used to reduce the number of subsequent computational parameters. The third module contains a layer of LSTM with 128 units (or Bi-LSTM to make the model non-causal), which is used to implement the function of temporal conclusion. As the modeling from the study, each eating process contains a series of ordered micromovements, such as picking up food, delivering food to mouth, idle state, putting down the hands, etc (Kyritsis et al., 2019). Thus, the orders of the output and the input are correlated. The fourth module has only one neuron as the full connection layer (FC layer), which is used to combine the outputs of all the channels and select the sigmoid as the activation function for probability mapping. A dropout rate of 0.05 was set prior to FC layer; binary cross entropy was used as loss function, with epoch set to 40, batch size set to 800; learning rate was set to 0.001 and $RMSprop$ was used as the optimizer.

4.5.3 Post Processing

In the post processing scheme, the first moment within each segment is marked as the time of the corresponding output segment. An example of the output segment from the model is shown in Figure 4.9 (the blue curve shows the original output; the red curve shows the output after thresholding processing with a threshold of 0.5). Two major problems can be spotted, being the existence of sharp noises and sharp cracks. Therefore, three further post processing steps were taken: Gaussian filtering, thresholding and length filtering. A kernel size of 9 was selected for the Gaussian filter. With the sampling frequency of 64Hz of the IMU and the sliding stride of 20, the coverage of the Gaussian kernel can be derived as follows:

$$\text{Coverage} = \frac{\text{stride} * \text{kernelSize}}{\text{samplingRate}} = \frac{20*9}{64} \approx 3s \quad (4.1)$$

According to the Gaussian distribution, if the kernel size is 6 times of σ, the Gaussian filter can achieve desirable results, when the sum of factors in the kernel is approximated to 1. Thus, σ was set to 1.5, kernel size was set to 9. The results after filtering and thresholding processing is shown in Figure 4.9, where sharp noises and sharp

cracks have been eliminated. The final prediction was obtained by filtering out the results with prediction time less than 1 second by length filtering.

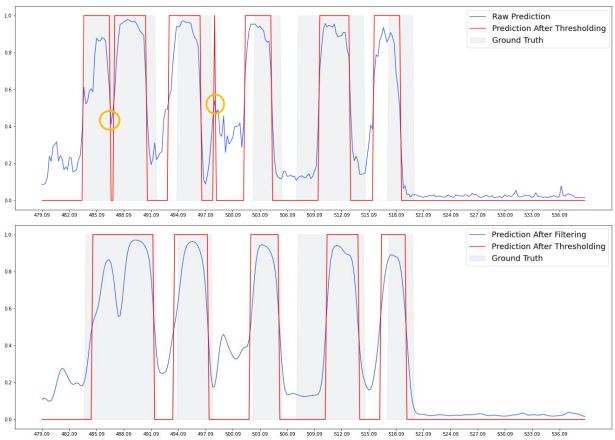

Figure 4.9: Results before and after filtering and thresholding

4.6 CNN-LSTM-Attention

The preprocessing and post processing schemes of CNN-LSTM-Attention (model 4.6) are identical to those of CNN-LSTM (model 4.5), and thus can be found in section 4.5.1 and 4.5.3. As shown in Figure 4.10, model 4.6 shares similar architecture to model 4.5, yet with the setting of $returnSequence = True$ in the LSTM. Thus, for each channel, the result of each hidden state at each timestamp will be weighted and summed by the attention model to obtain one single output. The advantage of such a setting is to take into account the state of each timestamp in the process, rather than only considering the final output. The corresponding results and discussions can be found in subsequent chapters.

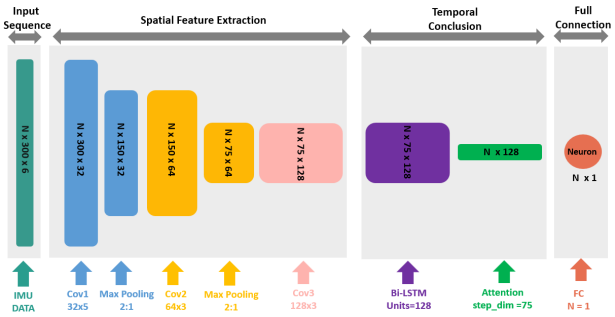

Figure 4.10: CNN-LSTM-Attention Architecture

4.7 TCN

4.7.1 Preprocessing

Since the IMU data is too long to be input into the model at once, the original data was down-sampled. The advantage of down sampling was that the input segment could contain information for a longer period of time without extending the length of the input sequence. As for labeling, point-wise annotations were made for each moment, rather than the segment-wise annotation mentioned in section 4.5.1, where the middle point was selected to represent the entire segment. For training datasets, the segment length was set to 2000, the down-sample factor was set to 3 and the sliding stride was set to 2000, so as to guarantee overlaps between segments, achieving data augmentation. As such, each input segment contained information:

$$L_{\text{info}} = \frac{2000 * 3}{64Hz} \approx 94s \tag{4.2}$$

For the validation set, the sliding stride was set to 6000, so as to guarantee the temporal continuity of the segment.

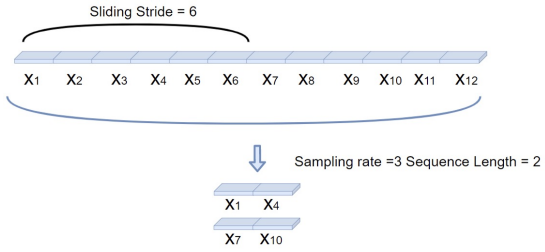

Figure 4.11: TCN Preprocessing

4.7.2 Architecture

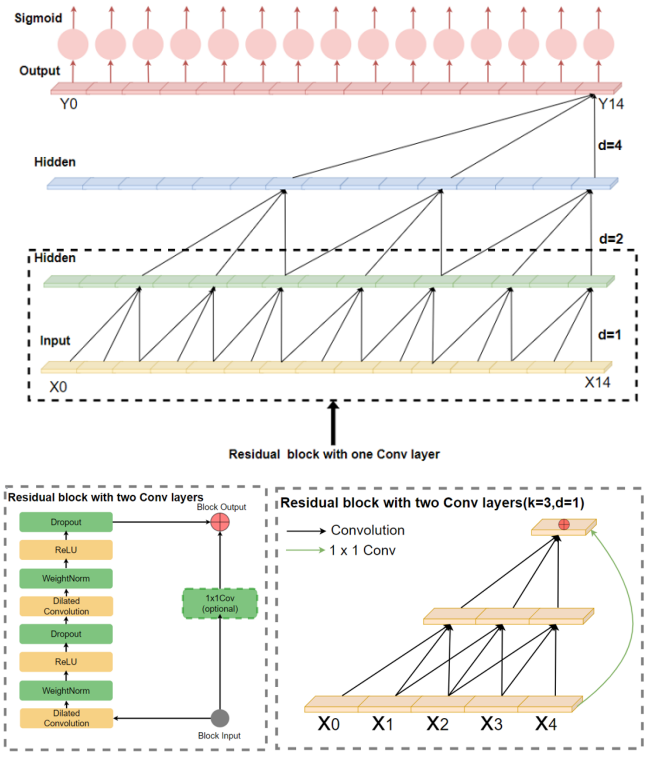

Figure 4.12: TCN Architecture

The fundamental structure of TCN (casual version) is shown in Figure 4.12 (Bai et al., 2018). Unlike CNN whose basic unit is a layer, the basic unit of TCN is a residual block. Each block contains two convolution layers with the same dilation factor (for simplicity, only one layer was shown in Figure 4.12), some normalization layers and activation functions. The input and output of each block are processed through a residual connection to facilitate the gradient flow. Besides, there is a $1*1$ convolution operation to ensure that the input and output have the same number of channels to be added directly. Different blocks have different dilation factors, which increase exponentially to further expand the receptive field. The receptive field is computed according to Equation 4.3. All blocks have the same number of filters to keep the number of channels constant. The output of the last block passes through a sigmoid layer for probability mapping. The hyper-parameters were eventually set as follows:

A dropout rate of 0.05 set prior to FC layer, number of filters set to 64, kernel size set to 3; binary cross entropy used as loss function, with epoch set to 70 and batch size set to 200, learning rate set to 0.0005 and $RMSprop$ used as the optimizer; 9 residual blocks used with dilation factor exponentially increasing from 1 to 256.

Noncausal TCN model with the same hyper-parameters was also tested. The corresponding results and discussions can be found in subsequent chapters.

$$R_{\text{field}} = 1 + 2 \cdot (K_{\text{size}} - 1) \cdot \sum_i d_i \tag{4.3}$$

4.7.3 Post Processing

For the TCN model, the same post-processing techniques as section 4.5.3 were used: Gaussian filtering, thresholding and length filtering. However, the Gaussian kernel size was set to 60, down-sample factor was 3, the kernel coverage was calculated as Equation 4.4. Accordingly, σ was set to 10, and the threshold of length filtering was set to 1 second.

$$\begin{aligned} \text{Coverage} &= \frac{\text{kernelSize} * \text{down-sampleFactor}}{\text{IMU samplingRate}} \\ &= \frac{60*3}{64Hz} = 2.8125s \approx 3s \end{aligned} \tag{4.4}$$

4.8 MS-TCN

MS-TCN is called Multiple-Stage Temporal Convolution Network (Farha & Gall, 2019). The basic unit of MS-TCN is a dilated residual layer (DR-layer), as shown in Figure 4.13. Each DR-layer contains a dilated convolution layer, a ReLU activation layer and a $1*1$ convolution layer to integrate some inter-channel relationships. Finally, a residual connection is used between layer input and output to facilitate the gradient flow.

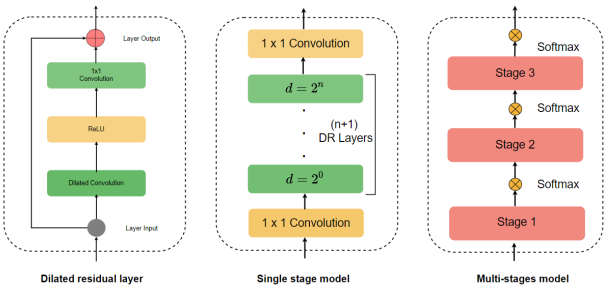

Figure 4.13: MS-TCN Architecture

Multiple DR-layers can form a single-stage model (SS-model). A SS-model starts with a $1*1$ convolution layer to perform channel mapping for the original input signal, followed by a series of dilated convolution layers with exponentially increasing dilation factors. It ends with another $1*1$ convolution layer used for channel mapping to make output in line with the class number $[0,1,0]$. While the number of filters in the last $1*1$ convolution layer equals the number of classes, all other convolution layers share the same number of filters.

Multiple SS-models can form a multiple-stage model (MS-model). Each layer of SS-model will output an initial prediction after Softmax activation. Then the output will be passed to the next layer of SS-model as new input. The last output of the SS-model will be used as the final prediction of the model. The corresponding receptive field for noncausal model can be calculated as follows:

$$\text{ReceptiveField}\ (l) = 2^{l+1} - 1 \tag{4.5}$$

where L represents the number of DR-layers in each SS-model. The loss function is made up of two parts: cross entropy loss, which is used for the classification problem:

$$\mathcal{L}_{cls} = \frac{1}{T} \sum_t -\log\left(y_{t,c}\right) \tag{4.6}$$

truncated mean square error (MSE), which is used to make the results of adjacent points as coherent as possible to avoid over-segmentation:

$$\begin{aligned} \mathcal{L}_{T-MSE} &= \frac{1}{TC} \sum_{t,c} \tilde{\Delta}_{t,c}^2 \\ \tilde{\Delta}_{t,c} &= \begin{cases} \Delta_{t,c} & : \Delta_{t,c} \leq \tau \\ \tau & : \text{otherwise} \end{cases} \\ \Delta_{t,c} &= |\log y_{t,c} - \log y_{t-1,c}| \end{aligned} \tag{4.7}$$

The final loss function is a combination of the above mentioned losses:

$$\mathcal{L}_s = \mathcal{L}_{cls} + \lambda \mathcal{L}_{T-MSE} \tag{4.8}$$

The hyper-parameters of this model were eventually set as follows:, $\lambda = 0.15$, $\tau = 4$, $numLayer = 10$, $numStage = 2$, $kernelSize = 3$, $learningRate = 0.0005$, $batchSize = 200$, $epoch = 70$, $RMSprop$ was used as the optimizer, model is set to non-causal mode.

The post processing scheme of MS-TCN (model 4.8) is identical to those of CNN-LSTM (model 4.5) and TCN (model 4.7), and thus can be found in section 4.7.1 and 4.7.3.

4.9 Evaluation Criteria

For meal session detection in full-day dataset, IoU score was used for evaluation. For bite detection in in-meal dataset, there are two types of evaluation criteria, focusing on point-wise and segment-wise evaluation, respectively.

The point-wise metrics, such as precision, recall and F1 score, evaluate the overall detection results for each point in detail. However, the point-wise metrics can not clearly present the performance of segment-wise detection, including the over-segmentation problems caused by noise, or a long segment misdetected as two separate segments.

In this study, the segment-wise metric $F1@k$ is adopted (Lea et al., 2017). For a certain point in a temporal sequence, it is considered to be positive when its corresponding probability value (P-value) is above P-threshold. Then, a segment with consecutive positive points is compared with the ground truth to calculate IoU score. The segment is considered to be true positive only if its IoU score is greater than IoU-threshold@k, where k is the IoU-threshold value. The precision, recall and F1 score of the segment will be further calculated.

The $F1@k$ metric is capable of combining the detection performance of the point-wise metrics and the segmentation performance of the segment-wise metrics. In this study, with P-threshold value set to 0.5, the performances were tested under IoU-threshold values of 0.1, 0.25 and 0.5.

4.10 Meal Session Detection

The main goal of meal session detection is to determine the temporal range of each meal session within a day and further detect all the bites, achieving fine-grain detection. In this part of the study, MS-TCN was selected as the prediction model for its overall better performance. Besides, the loss function of MS-TCN includes truncated MSE to minimize the over-segmentation problem, which is essential for free living detection. The preprocessing, post processing and all hyper-parameters were set consistent with model 4.8. Since the same timestamps were necessary for both left-hand and right-hand data for range detection, the previously used offset-appending scheme could not be used to integrate the two sets of data. Therefore, the left-hand dataset $Full - Day - L$ and right-hand dataset $Full - Day - R$ were input respectively. An OR operation on each corresponding timestamp in both dataset was adopted to fuse the two sets of data, as shown in Figure 4.14.

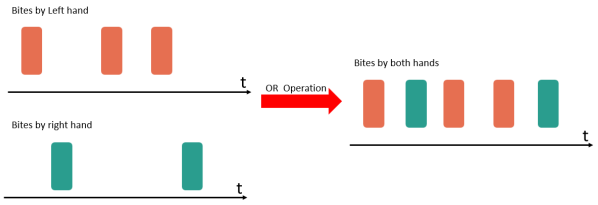

Figure 4.14: OR Operation for Two-handed Datasets

An example of processed detection results of $Full - Day$ dataset is shown in Figure 4.16, where the predicted bites by the model are represented in red. Following were two operations on the initial prediction: Clustering by Morphology Closing and Density Filtering.

With morphology closing, all adjacent bites were clustered to an interval as shown by the green line, representing the meal-time range. The step size of the closing was set to 2500, thus covering a period of 4minutes, which was calculated as Figure 4.15:

$$\begin{aligned} \text{Coverage} &= \frac{\text{stepSize} * 2 * \text{down-sampleFactor}}{\text{IMU SamplingRate}} \\ &= \frac{2500 * 2 * 3}{64Hz} = 234s \approx 4 \text{ mins} \end{aligned} \tag{4.9}$$

Some of the intervals were caused by noise, and thus density filtering was necessary. The total length of bites in each interval was calculated and divided by the length of the interval to obtain the bite density for density filtering. The filtering rule is shown in Figure 4.15.

With both morphology closing and density filtering, two intervals were obtained as shown in Figure 4.16: meal 1 with bite density of 0.25 and meal 2 with bite density of 0.31. The IoU score based on the ground truth was then obtained for each interval: IoU score of 0.99 for meal 1, and 0.85 for meal 2. After the meal-time ranges were obtained, in order to achieve finer-grain detection, all bites

```
if mealtime length <300s:
      Filtered out
else if mealtime length = [300s,600s):
    if density < 10%:
      Filtered out
else if mealtime length = [600s,1000s):
    if density < 5%:
      Filtered out
else if mealtime length >1000s:
    if density < 3%:
      Filtered out
```

Figure 4.15: Density Filtering Algorithm

within the meal-time ranges were selected and compared with the annotations (bites based on ground truth) to calculate the $F1@10$ score, which was 90% for meal 1 and 88% for meal 2, respectively.

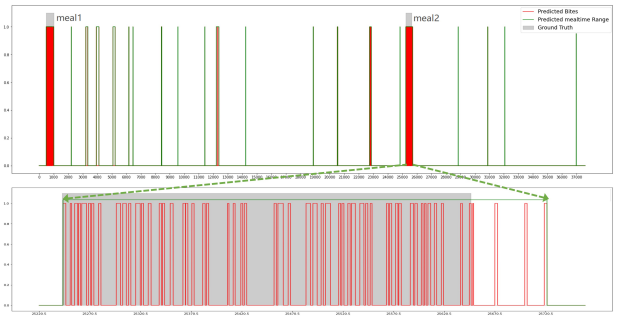

Figure 4.16: Results of the Example Free Living Detection

5 EXPERIMENTS AND RESULTS

5.1 LOSO Validation Based on Each Meal for In-Meal Dataset

In this section, each meal session was used as the validation dataset to be tested in LOSO validation with multiple models. Results were compared to evaluate the performance of different models. The best performance of bite detection (evaluated by $F1@10$) was observed in the MS-TCN model (91.3%), higher than that of CNN-LSTM (88.4%). The best performance of bite segmentation (evaluated by $F1@50$) was also observed in the MS-TCN model (87.7%), much higher than that of CNN-LSTM (36.7%). A detailed discussion will be drawn in section 6.4.

	TP @10,25,50	FP @10,25,50	FN @10,25,50	F1 @10,25,50(%)
CNN-LSTM	2197, 2047, 972	288, 438, 1513	404, 554, 1629	86.4, 80.5, 38.2
CNN-BiLSTM	2243, 2092, 931	232, 383, 1544	358, 509, 1670	88.4, 82.4, 36.7
CNN-BiLSTM (Attention)	2193, 2052, 851	278, 419, 1620	408, 549, 1750	86.5, 80.9, 33.6
TCN (Causal)	2145, 2140, 1931	264, 269, 478	456, 461, 670	85.6, 85.4, 77.1
TCN (Non-Causal)	2316, 2316, 2214	216, 216, 318	285, 285, 387	90.2, 90.2, 86.2
MSTCN (Non-Causal)	2378, 2377, 2286	232, 233, 324	223, 224, 315	**91.3**, 91.2, 87.7

Table 5.1: LOSO validation of meal dataset on different models

5.2 LOSO Validation Based on Each Subject for In-Meal Dataset

In this section, the meal sessions of each subject (including two or four meal sessions) were used as the validation dataset to be tested in LOSO validation with multiple models. Results were compared to reveal the influence of individual characteristics, e.g. eating habits of different subjects.

MSTCN	k=10	k=25	k=50
TP	2271	2267	2150
FP	307	311	428
FN	330	334	451
F1	87.7%	87.5%	83.1%

Table 5.2: LOSO validation by subject

5.3 LOSO Validation for Full-Day Dataset

In this section, for validating each long-term (full-day) dataset, all meal-session data from this day (including one or two meal sessions) were used as the validation dataset to be tested in LOSO validation with MS-TCN model. The results are shown in Figure 5.1. Results were observed to the performance of the MS-TCN model to segment the long-term data to a given temporal range of a meal session (IoU score of 0.91), as well as the performance of fine-grain detection for each bite ($F1@10$ score of 88.1%). The results are shown as Table 5.3:

MSTCN	k=10	k=25	k=50
TP	2238	2131	1621
FP	342	449	959
FN	263	370	880
F1	88.1%	83.9%	63.8%
Intersection		40850s	
Union		44954s	
Mean IoU		0.91	

Table 5.3: LOSO validation for free-living dataset

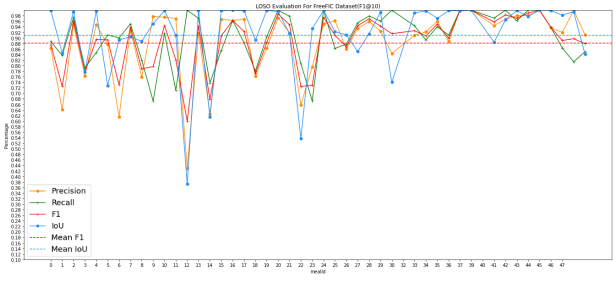

Figure 5.1: Result for LOSO Validation for Full-Day Dataset

5.4 Public Meal Dataset Comparison

In this section, the publicly available Food Intake Cycle (FIC) Dataset was used for verification (Kyritsis et al., 2021). In the FIC dataset, a total of 12 subjects were

recorded during their meal sessions. The total duration of the 21 meals sums up to 246 minutes, with a mean duration of 11.7 minutes. All data were measured by a 6-DoF IMU sensor on one wrist. The performance of the LSTM model was used as a benchmark so as to compare the performance of the proposed model (MS-TCN). The hyper-parameters were set identical to those of model 4.5.

	TP @10,25,50	FP @10,25,50	FN @10,25,50	F1 @10,25,50(%)
LSTM	1178, 1153, 885	259, 284, 552	154, 179, 447	85.1, 83.3, 63.9
MSTCN	1138, 1125, 987	106, 119, 257	194, 207, 345	88.4, 85.9, 76.6

Table 5.4: Public dataset verification

6 DISCUSSION

6.1 Remarks on Annotation and Collected Data

The data used in this study are primarily collected with experiments. Due to the uncertainty of human behavior, there was bound to be variance in the behavior of the subjects (Selamat & Ali, 2020).

A common situation is that the subject collected the food on the fork well in advance to the bite. Figure 6.1 shows the actions of a subject during a bite. A piece of food has been placed idle on the fork for about 15 seconds. In such cases, only the part that contains more information of wrist movement is marked as a bite, which usually starts from the movement of the wrist to send food into the mouth, and ends with putting down the wrist (Stankoski et al., 2021).

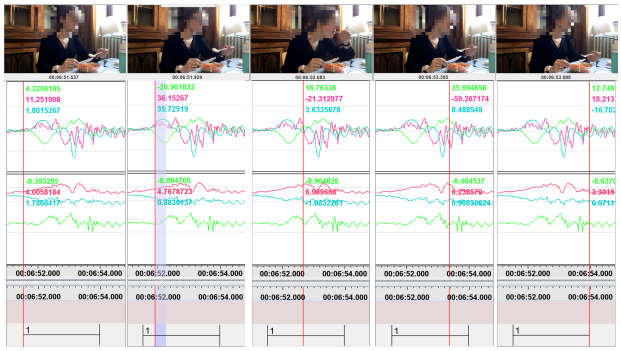

Figure 6.1: Waveform and Video Recording of Subject 33

In short, each annotation is desired to include hand rising, food delivery into the mouth and hand lowering movements. But in operations, a bite could only include the latter two movements. In comparison, Figure 6.2 shows a full bite with all four movements within 3 seconds.

When the waveform of the inertial data of each bite was observed, it was apparent that some eating behaviors led to a clear pattern of waveform (Dong et al., 2012), while

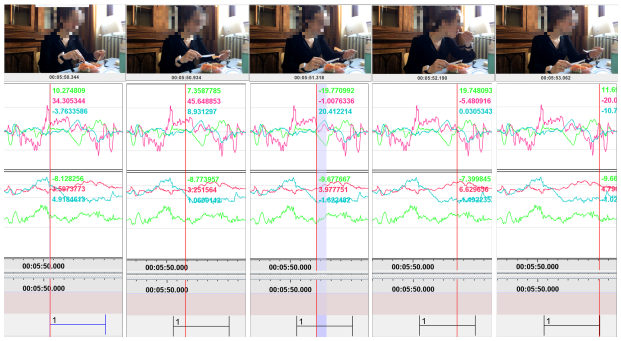

Figure 6.2: Waveform and Video Recording of Subject 33

some eating behaviors led to poor inertial data. The subject in Figure 6.3 was holding the fork in front of the mouth. When he was prepared for another bite, he moved forward his head while slightly spinning his wrist to deliver the food into his mouth. As such, little movement of the wrist was detected by the wristband, and the corresponding data could result in a missing positive point in bite detection. Such behaviors occur mostly when the subject has the habit of picking up and holding the food before the mouth prior to the next bite, and when the subject eats noodles or food in larger pieces, for which the subject is more likely to lower the head instead of raising the wrist to deliver food into the mouth. These problems were mainly observed among the first few subjects.

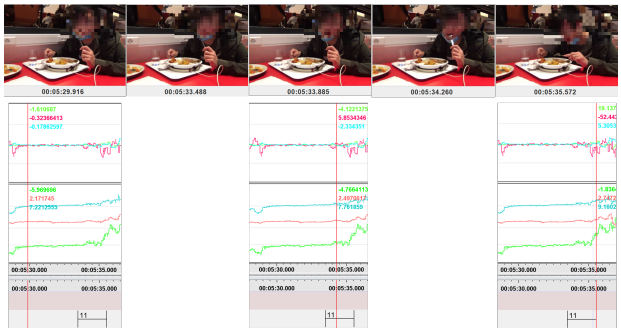

Figure 6.3: Waveform and Video Recording of Subject 7

6.2 Influence of Hyper-parameters

In the CNN-LSTM model, the setting of hyper-parameters referred to the related research by Kyritsis et al. (2021), which is used as a benchmark to compare the subsequent models. The overall model architecture is based on a single convolution layer, rather than pairs of convolution layers. By using $32 * 5$, $64 * 3$ and $128 * 3$ VGG architecture in the convolution part, the extraction from wide to fine in space (kernel size from 5 to 3) and from simple to complex in features (filter size from 32 to 128) was realized (Simonyan & Zisserman, 2014). Generally speaking, the larger the number of filters, the more complex the extracted features, the better the model results. The ReLU

function was used as the activation function except for the fully connected layer, where sigmoid was adopted for probability mapping. The purpose of the ReLU function was to promote gradient flow.

For TCN models, the use of skip connection usually accelerates the training process, mainly due to the promotion of gradient flow. Dropout usually has a slight effect and thus can be given a small value. Although the results of the model usually get better as the number of filters increases, too many filters not only increase the training parameters and thus the training time, but also cause overfitting problems in case of inadequate data. Similarly, the more layers of dilation, the better the model will look, but the model parameters will also increase significantly. It is usually sufficient to ensure that the corresponding receptive field can cover the sequence length. Regarding normalization, unless the model has many parameters or little training data, it usually has little effect on the performance of the model. Besides, the introduction of normalization would slow down the training speed. Therefore, normalization was not applied in this model. In the training process, the most critical parameter was the kernel initializer. Initially, the kernel initializer was set as glorot-uniform, with which the gradient could not descend. By setting it to random-normal, training was carried out quickly. Thus, in case that the gradient would not drop, it is worth trying to change the kernel initializer.

For MS-TCN models, the multi-stage model showed better performance than the single-stage model, with the same number of parameters. The former also led to less over-segmentation errors (Farha & Gall, 2019). However, increasing stages would significantly increase the difficulty of training and might lead to overfitting, which would ultimately affect model accuracy. Therefore, the two-stage model was adopted. Similar to TCN, the higher the number of layers, the better the final performance of MS-TCN would be, but the corresponding training difficulty would also increase. It is usually sufficient to ensure that the corresponding receptive field can cover the sequence length. The number of filters did not have a significant impact on the final result, though it should be properly configured according to the complexity of the research problem and the input data. Regarding the settings of λ and τ, the larger the value of these two over-segments, the greater the weight of truncated MSE loss on the overall loss function, the closer the output values of successive points of the model, and the lower the probability of over-segmentation. As such, more positive points might not be predicted as true positives. Regarding the sampling rate in segment generation, although longer sequences could be obtained with the same input sequence length while increasing the sampling rate, resolution loss occurs accordingly. The parameters were thus set as follows: down-sample factor of 3, segment length of 2000, which means a sequence of length 6000 could be obtained.

6.3 Analysis on Free-Living Detection

Since the step size of morphological closing was set to 2500, all bites within an interval of 234 seconds would be integrated. While the duration of each meal session varied, the longer the meal session, the smaller the meal density. Thus, all meal sessions were categorized into four types by duration. As shown in Figure 5.1, intervals of bites could be segmented with high IoU scores for all 46 meal sessions (meal number 31 and 38 were not associated with a full-day experiment). The meal duration was defined from the beginning of the first bite to the end of the last bite, rather than the length of the corresponding video recording. Even though some meal sessions did not obtain a high IoU score, the final $F1@10$ value was adequately high. It was observed that at the beginning and end of the meal, there were more interfering actions and the subjects were not fully in the stage of food consumption, e.g. placing items after a bite, or fetching items from the bag before a bite. Thus, some bites in these parts could not be detected accurately by the model, while the denser bites in the middle of the meal were more likely to be detected accurately.

Compared to the long-time detection method based on Gaussian filtering (Kyritsis et al., 2021), three advantages could be observed in the free-living detection model in this study: First, it achieved better generalization performance, where all detection of meal-session range was based on the same criteria (step size, density filtering). With Gaussian filters, it is difficult to select the threshold after filtering for different duration and meal densities. Second, clustering could be applied without changing the original results. With Gaussian filters, even if edge detectors with side-lobes were adopted, the left and right boundaries would still be damaged. Third, by implementing OR operation on two-handed data, the corresponding meal-session range could still be detected even if the subject kept switching hands during the meal.

At the same time, some disadvantages still exist in this approach. First, an additional interval of bites was detected (false positive) in $Full-Day$ dataset ID 14, which could not be filtered out, as shown in Figure 6.4. The length of this interval was 360 seconds, with a density of 0.14, which qualified for the evaluation standards. It was hypothesized that this interval was possibly caused by out-of-meal snack intake. Since additional out-of-meal food intake activities were not considered during experiment design and data collection, the hypothesis could not be verified. Second,

the mean $F1@10$ value was calculated 0.88, lower than the result 0.91 in the LOSO validation by each meal, for the following reasons. One reason is that the use of OR operation on two-handed data would destroy the independence of data, e.g. noise recorded on the right hand would have an impact on the positive prediction of the left hand. Another reason is that in the LOSO validation of $Full-Day$ dataset, all the meal-session data within the corresponding full-day data were used as validation dataset, while for most of the collected data, only two meals of that same day were collected for each subject. In other words, for the majority of the subjects, in each LOSO, both meals from each of them were used as validation rather than training (2 meals in total, including 1 or 2 meals for validation). Therefore, some unique meal intake habits about individuals could not be learned by the model. To verify this hypothesis, the experiment 5.2 was designed, where LOSO validation was performed using the all meal-session data of each subject as validation (2 or 4 meals in total, all meals used for validation). The latter results were lower compared to the former. The most effective solution to this problem is to collect an adequately large amount of data to fully train the model. In summary, the detection of out-of-meal (snack) detection and the larger number of sample collections should be part of the future work.

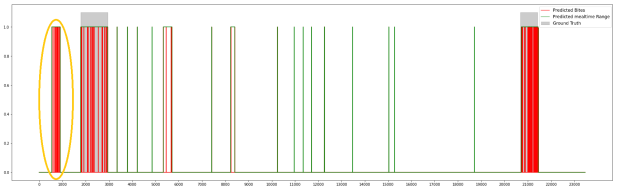

Figure 6.4: Results for Full-Day Dataset ID=14

6.4 Comparison of Different Models

6.4.1 Models Based on CNN-(Bi)LSTM Architecture

For the three models based on CNN-(Bi)LSTM architecture, the best performer is CNN-BiLSTM. Compared with LSTM, this architecture can achieve noncausal input and obtain better performance. All three models did not perform well at $F1@50$ score, potentially due to the following reasons:

First, the segmentation window needs to be set manually, while the duration of actions varies significantly. According to the output of the model, the prediction curve shows a trend of normal distribution. In a realistic sense, the model cannot predict the beginning and end of the bite accurately, mainly due to lack of clear characteristics in actions. The use of artificially segmented sequences would destroy the integrity of the data, so that the edge points may not be correlated with the strong features in the middle part of the bite.

Second, RNN models were proved poor at long-distance memory, despite their theoretically infinite memory capacity (Bai et al., 2018). Moreover, in RNN models, previous states for every timestamp are mixed with the following timestamps, which leads to the introduction of Attention architecture, aiming to achieve a direct correlation between output and input, instead of final temporal states based on LSTM architecture. However, the results of the experiment did not display improvement in performance, which could be due to the short length of the input sequence, from which the information extracted was limited and incomplete.

6.4.2 Models Based on Temporal Convolution Network Architecture

For architectures based on temporal convolution networks, not only have the $F1@10$ and $F1@25$ scores increased, but the $F1@50$ score has stepped up significantly, potentially due to the following reasons:

First, the input to TCN is the original data points. No manual segmentation was required to create segmentation windows which would destroy the integrity of bites. The input sequence is also long enough for the model to learn a wider range of information (e.g. for an input sequence of length 2000 and downsample factor 3, in noncausal mode, each input could obtain 6000 information points).

Second, the TCN model based on dilated convolution can extract the feature of longer, discrete and direct temporal relation. It can also avoid the recursive hidden states compared to LSTM architecture. TCN has been shown to have superior detection performance over LSTM on several experiments, and should be the first model considered for similar studies (Bai et al., 2018). This argument is further demonstrated in this study, and it is concluded that TCN has even more outstanding advantages in segmentation performance. For MSTCN, it is motivated by the idea of stacked predictors and this architecture has been shown significant improvement than single layer predictor (Wei et al., 2016)(Newell et al., 2016). More importantly, its loss function includes truncate MSE loss, which can avoid over segmentation as much as possible and is vital for the Full-Day data used in this study. It can be seen from Table 5.1 that MS-TCN achieves the best results among the six models in terms of both detection performance and segmentation performance.

6.5 Discussion of Public Dataset

Public datasets were also used in multiple stages of this study. The LSTM is used as a benchmark to compare with the MS-TCN proposed in this paper. The results are shown as Table 5.4, which also proves the superior performance of MS-TCN in detection and segmentation. However, it was found that the difference in performances of the two models on $F1@50$ metrics was much smaller than our dataset. Thus, statistics on both datasets were made to obtain the density of the all meal sessions and the average length of bite for each, as shown in Table 6.1:

	Average Bite Length	Average Bite Density
Experiment Dataset	2.3420 seconds	0.0629
Public Dataset	4.5218 seconds	0.2503

Table 6.1: Comparison in Average Bite Length and Bite Density

It can be observed that the average length and density of bites in the public dataset are much larger than our dataset, especially the density of bites. The possible reasons are as follows. First, our dataset consists of two-handed data. With offset appending operation, the total length of all meal sessions doubled, resulting in density halfing. Second, the annotation of our dataset did not include food pick-up action, since it was observed that pick-up actions had too much variation and thus not much regularity. Thus, if the pick-up actions were included in the bite cycle, it may interfere with the model training. Therefore, the high bite density in the public dataset might lead to the bias in model prediction and tendency in true prediction, resulting in higher $F1@50$ score of the public dataset.

7 CONCLUSION

In this paper, an approach to perform fine-grain bite detection based on a full-day free living dataset is proposed. It not only achieves segmentation of meal session intervals out of a full-day dataset, obtaining an average IoU score of 0.9, but also succeeds in the validation of bite detection, scoring 0.88 in average F1 score. Initially, by comparing different models using $F1@10$ and $F1@50$ scores, MS-TCN was found to have the best detection performance (evaluated by $F1@10$ score) and performs much better in segmentation than the traditional LSTM architecture (evaluated by $F1@50$ score). Next, it was observed that there are certain patterns in the eating habits of different individuals. With LOSO validation by each subject, two types of patterns were discovered: general patterns that can be learned from other subjects, and personal patterns that only belongs to one specific subject, only existing in his or her personal habits. Additionally, a public dataset used to further evaluate the MS-TCN model proposed in this study, using LSTM as a benchmark. Results showed that even on the public dataset, the proposed MS-TCN model still performs better than LSTM. Future works are needed to improve experiment design with out-of-meal food intake (snack) behavior and collect a larger scale of data.

8 ACKNOWLEDGEMENTS

This thesis project would not have succeeded without the assistance of many people. We would like to thank our supervisors Prof. Bart Vanrumste and Prof. Hans Hallez, as well as our co-supervisor Chunzhuo Wang, for their generous guidance throughout this project. In addition, we are indebted to all the subjects who have participated in the experiment of the study, without whom we would not have obtained a necessary dataset. Lastly, our family and friends deserves our ultimate gratitude. Thanks to their support, we could accomplish the final stage of this thesis.

BIBLIOGRAPHY

Anthimopoulos, M., Dehais, J., Shevchik, S., Ransford, B. H., Duke, D., Diem, P., & Mougiakakou, S. (2015). Computer vision-based carbohydrate estimation for type 1 patients with diabetes using smartphones. *Journal of Diabetes Science and Technology*, *9*, 507–515. https://doi.org/10.1177/1932296815580159

Anthimopoulos, M. M., Gianola, L., Scarnato, L., Diem, P., & Mougiakakou, S. G. (2014). A food recognition system for diabetic patients based on an optimized bag-of-features model. *IEEE Journal of Biomedical and Health Informatics*, *18*, 1261–1271. https://doi.org/10.1109/jbhi.2014.2308928

Bai, S., Zico, K. J., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv.org*. Retrieved November 15, 2019, from https://arxiv.org/abs/1803.01271

Consensyspro software. (n.d.). *Shimmer Wearable Sensor Technology*. Retrieved May 9, 2022, from https://shimmersensing.com/product/consensyspro-software/

Crispim, C. A., Zalcman, I., DÃ¡ttilo, M., Padilha, H. G., Edwards, B., Waterhouse, J., Tufik, S., & Mello, M. T. d. (2007). The influence of sleep and sleep loss upon food intake and metabolism. *Nutrition Research Reviews*, *20*, 195â212. https://doi.org/10.1017/S0954422407810651

Dong, Y. (2012). Tracking wrist motion to detect and measure the eating intake of free-living humans. Retrieved May 17, 2022, from http://cecas.clemson.edu/~ahoover/theses/dong-diss.pdf

Dong, Y., Hoover, A., Scisco, J., & Muth, E. (2012). A new method for measuring meal intake in humans via automated wrist motion tracking. *Applied Psychophysiology and Biofeedback*, *37*, 205–215. https://doi.org/10.1007/s10484-012-9194-1

Dongwoo, K., & Kim, H. C. (2007). Activity energy expenditure assessment system based on activity classification using multi-site triaxial accelerometers. *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. https://doi.org/10.1109/iembs.2007.4352781

Download elan 6.3.0. (2022). *softpedia*. Retrieved May 9, 2022, from https://www.softpedia.com/get/Multimedia/Video/Other-VIDEO-Tools/ELAN.shtml

Farha, Y. A., & Gall, J. (2019). Ms-tcn: Multi-stage temporal convolutional network for action segmentation. *arXiv:1903.01945 [cs]*. Retrieved November 3, 2021, from https://arxiv.org/abs/1903.01945

Farooq, M., & Sazonov, E. (2016a). Automatic measurement of chew count and chewing rate during food intake. *Electronics*, *5*, 62. https://doi.org/10.3390/electronics5040062

Farooq, M., & Sazonov, E. (2016b). A novel wearable device for food intake and physical activity recognition. *Sensors*, *16*, 1067. https://doi.org/10.3390/s16071067

Farooq, M., & Sazonov, E. (2017). Segmentation and characterization of chewing bouts by monitoring temporalis muscle using smart glasses with piezoelectric sensor. *IEEE Journal of Biomedical and Health Informatics*, *21*, 1495–1503. https://doi.org/10.1109/jbhi.2016.2640142

Fukuike, C., Kodama, N., Manda, Y., Hashimoto, Y., Sugimoto, K., Hirata, A., Pan, Q., Maeda, N., & Minagi, S. (2014). A novel automated detection system for swallowing sounds during eating and speech under everyday conditions. *Journal of Oral Rehabilitation*, *42*, 340–347. https://doi.org/10.1111/joor.12264

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/cvpr.2016.90

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kang, D. W., Choi, J. S., Lee, J. W., Chung, S. C., Park, S. J., & Tack, G. R. (2010). Real-time elderly activity monitoring system based on a tri-axial accelerometer. *Disability and Rehabilitation: Assistive Technology*, *5*, 247–253. https://doi.org/10.3109/17483101003718112

Khazrai, Y. M., Defeudis, G., & Pozzilli, P. (2014). Effect of diet on type 2 diabetes mellitus: A review. *Diabetes/Metabolism Research and Reviews*, *30*, 24–33. https://doi.org/10.1002/dmrr.2515

Kyritsis, K., Diou, C., & Delopoulos, A. (2019). Modeling wrist micromovements to measure in-meal eating behavior from inertial sensor data. *IEEE Journal of Biomedical and Health Informatics*, *23*, 2325–2334. https://doi.org/10.1109/jbhi.2019.2892011

Kyritsis, K., Diou, C., & Delopoulos, A. (2021). A data driven end-to-end approach for in-the-wild monitoring of eating behavior using smartwatches. *IEEE Journal of Biomedical and Health Informatics*, *25*, 22–34. https://doi.org/10.1109/jbhi.2020.2984907

Lansky, D., & Brownell, K. D. (1982). Estimates of food quantity and calories: Errors in self-report among obese patients. *The American Journal of Clinical Nutrition*, *35*, 727–732. https://doi.org/10.1093/ajcn/35.4.727

Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2017.113

Lin, C.-W., Yang, Y.-T. C., Wang, J.-S., & Yang, Y.-C. (2012). A wearable sensor module with a neural-network-based activity classification algorithm for daily energy expenditure estimation. *IEEE Transactions on Information Technology in Biomedicine*, *16*, 991–998. https://doi.org/10.1109/titb.2012.2206602

Lopez-Meyer, P., Schuckers, S., Makeyev, O., & Sazonov, E. (2010). Detection of periods of food intake using support vector machines. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society IEEE Engineering in Medicine and Biology Society. Annual International Conference*, *2010*, 1004â1007. https://doi.org/10.1109/iembs.2010.5627796

Makeyev, O., Lopez-Meyer, P., Schuckers, S., Besio, W., & Sazonov, E. (2012). Automatic food intake detection based on swallowing sounds. *Biomedical Signal Processing and Control*, *7*, 649–656. https://doi.org/10.1016/j.bspc.2012.03.005

Matlab - mathworks. (n.d.). *www.mathworks.com*. https://mathworks.com/products/matlab.html

Mattfeld, R. S., Muth, E. R., & Hoover, A. (2017). Measuring the consumption of individual solid and liquid bites using a table-embedded scale during unrestricted eating. *IEEE Journal of Biomedical and Health Informatics*, *21*, 1711–1718. https://doi.org/10.1109/jbhi.2016.2632621

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. *arXiv:1603.06937 [cs]*. Retrieved May 9, 2022, from https://arxiv.org/abs/1603.06937v2

Norberg, A., Athlin, E., & Winblad, B. (1987). A model for the assessment of eating problems in patients with parkinson's disease. *Journal of Advanced Nursing*, *12*, 473–481. https://doi.org/10.1111/j.1365-2648.1987.tb01356.x

Oddy, W. H., Robinson, M., Ambrosini, G. L., Oâ^2$Sullivan, T.A.$, de Klerk, N. H., Beilin, L. J., Silburn, S. R., Zubrick, S. R., & Stanley, F. J. (2009). The association between dietary patterns and mental health in early adolescence. *Preventive Medicine*, *49*, 39–44. https://doi.org/10.1016/j.ypmed.2009.05.009

Organization, W. H. (2016). Global ncd target: Halt the rise in obesity. *apps.who.int*. https://apps.who.int/iris/handle/10665/312281

Papapanagiotou, V., Diou, C., Ioakimidis, I., Sodersten, P., & Delopoulos, A. (2019). Automatic analysis of food intake and meal microstructure based on continuous weight measurements. *IEEE Journal of Biomedical and Health Informatics*, *23*, 893–902. https://doi.org/10.1109/jbhi.2018.2812243

Papapanagiotou, V., Diou, C., Langlet, B., Ioakimidis, I., & Delopoulos, A. (2015). A parametric probabilistic context-free grammar for food intake analysis based on continuous meal weight measurements. *2015 37th Annual*

International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). https://doi.org/10.1109/embc.2015.7320212

Pasler, S., & Fischer, W.-J. (2014). Food intake monitoring: Automated chew event detection in chewing sounds. *IEEE Journal of Biomedical and Health Informatics*, *18*, 278–289. https://doi.org/10.1109/jbhi.2013.2268663

Päßler, S., Wolff, M., & Fischer, W.-J. (2012). Food intake monitoring: An acoustical approach to automated food intake activity detection and classification of consumed food. *Physiological Measurement*, *33*, 1073–1093. https://doi.org/10.1088/0967-3334/33/6/1073

Raffel, C., & Ellis, D. P. W. (2016). Feed-forward networks with attention can solve some long-term memory problems. *arXiv:1512.08756 [cs]*. Retrieved May 9, 2022, from https://arxiv.org/abs/1512.08756

Rouast, P. V., Heydarian, H., Adam, M. T. P., & Rollo, M. E. (2020). Oreba: A dataset for objectively recognizing eating behavior and associated intake. *IEEE Access*, *8*, 181955â181963. https://doi.org/10.1109/ACCESS.2020.3026965

Schoeller, D. A. (1995). Limitations in the assessment of dietary energy intake by self-report. *Metabolism*, *44*, 18–22. https://doi.org/10.1016/0026-0495(95)90204-x

Selamat, N. A., & Ali, S. H. M. (2020). Automatic food intake monitoring based on chewing activity: A survey. *IEEE Access*, *8*, 48846–48869. https://doi.org/10.1109/access.2020.2978260

Sharma, S. (2020). Detecting periods of eating in everyday life by tracking wrist motion â what is a meal? *All Dissertations*. Retrieved May 17, 2022, from https://tigerprints.clemson.edu/all_dissertations/2675/

Sharma, S., & Hoover, A. (2022). Top-down detection of eating episodes by analyzing large windows of wrist motion using a convolutional neural network. *Bioengineering*, *9*, 70. https://doi.org/10.3390/bioengineering9020070

Sharma, S., Jasper, P., Muth, E., & Hoover, A. (2020). The impact of walking and resting on wrist motion for automated detection of meals. *ACM Transactions on Computing for Healthcare*, *1*, 1–19. https://doi.org/10.1145/3407623

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv.org*. https://arxiv.org/abs/1409.1556

Stankoski, S., Jordan, M., Gjoreski, H., & LuÅ¡trek, M. (2021). Smartwatch-based eating detection: Data selection for machine learning from imbalanced data with imperfect labels. *Sensors*, *21*, 1902. https://doi.org/10.3390/s21051902

Trovato, F. M., Martines, G. F., Brischetto, D., Catalano, D., Musumeci, G., & Trovato, G. M. (2015). Fatty liver disease and lifestyle in youngsters: Diet, food intake frequency, exercise, sleep shortage and fashion. *Liver International*, *36*, 427–433. https://doi.org/10.1111/liv.12957

Ulijaszek, S. J. (2003). Obesity: Preventing and managing the global epidemic. report of a who consultation. who technical report series 894. pp. 252. (world health organization, geneva, 2000.) sfr 56.00, isbn 92-4-120894-5, paperback. *Journal of Biosocial Science*, *35*, 624–625. https://doi.org/10.1017/s0021932003245508

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv.org*. https://arxiv.org/abs/1706.03762

Vincent, L. (1994). Morphological area openings and closings for grey-scale images. *Shape in Picture*, 197–208. https://doi.org/10.1007/978-3-662-03039-4_13

Wei, S.-E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. *arXiv:1602.00134 [cs]*. Retrieved May 17, 2020, from https://arxiv.org/abs/1602.00134

Yang, C.-C., & Hsu, Y.-L. (2009). Development of a wearable motion detector for telemonitoring and real-time identification of physical activity. *Telemedicine and e-Health*, *15*, 62–72. https://doi.org/10.1089/tmj.2008.0060

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv.org*. https://doi.org/10.48550/arXiv.1511.07122

Zhang, S., Nguyen, D., King, Z., Pradeep, J., & Alshurafa, N. (2018). Habits necklace. *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*. https://doi.org/10.1145/3267305.3267665

Zhang, W., Yu, Q., Siddiquie, B., Divakaran, A., & Sawhney, H. (2015). Âsnap-n-eatâ. *Journal of Diabetes Science and Technology*, *9*, 525–533. https://doi.org/10.1177/1932296815582222

Zhao, W., & Du, S. (2016). Spectralâspatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, *54*, 4544â4554. https://doi.org/10.1109/TGRS.2016.2543748