Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

Ahmed Isifan

Food intake monitoring using neural networks based on swallowing sound data

Master's thesis in Electronic Systems Design and Innovation
Supervisor: Dag Roar Hjelme
Co-supervisor: Salman Ijaz Siddiqui

June 2022

NTNU
Norwegian University of
Science and Technology

Ahmed Isifan

Food intake monitoring using neural networks based on swallowing sound data

Master's thesis in Electronic Systems Design and Innovation
Supervisor: Dag Roar Hjelme
Co-supervisor: Salman Ijaz Siddiqui
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Master Thesis
TFE4940 Electronic Systems Design and Innovation

Food intake monitoring using neural networks based on swallowing sound data

Ahmed Isifan

Supervisor:
Dag Roar Hjelme

Co-supervisor:
Salman Ijaz Siddiqui

Trondheim, June 13th,2022

NTNU
Norwegian University of
Science and Technology

Faculty of Information Technology and Electrical Engineering
DEPARTMENT OF ELECTRONIC SYSTEMS

Abstract

Artificial Pancreas systems rely on Continuous Glucose Monitoring systems to measure blood glucose levels and based on that measurement, the insulin dose is calculated and administered through an insulin pump. However, these systems are not perfect and suffer from time delays from the onset of the meal until the meal is detected and insulin is administered. Patients are required to announce their meal intake and also calibrate the system by taking a finger stick blood test on a daily basis. In a previous project, the work done by Konstanze k. on automatic meal onset detection using bowel sound recordings was continued, and swallowing sound recordings were introduced. This project continues the previous work and adapts such a system using only swallowing sound recordings. Furthermore, speech, meal onset, meal type, and swallowing detectors were built to monitor the food intake, this includes the duration of the meal, the meal type, and the number of swallows. This study showed the potential of using swallowing sound recordings to improve and work around the downsides of Artificial Pancreas systems.

For the first part of the project, 10 swallowing recordings were used to build a meal onset and a speech detector. Using these recordings, Power Spectral Density features were extracted and used for training and building Multilayer Perceptron classification models. The speech detector had an accuracy and an F1 score of 99%, while the meal onset detector had an accuracy and an F1 score of 95%.

For the second part of the project, 20 new recordings were acquired to build a meal type and a swallowing detector. These recordings were acquired using two pre-selected meals, oats and salad. The recordings were obtained using two subjects, where each recording was about 30 min to 35 min, and the microphone used for capturing the swallowing sounds was placed just above the collar bone on the neck. Using these recordings, Mel spectrogram features were extracted and used to build Convolutional Neural Network classification models. The swallowing detector had an accuracy of 93% and an F1 score of 92%, while the meal type detector had an accuracy and an F1 score of 96%.

Sammendrag

Kunstig Bukspyttkjertel systemer er avhengige av Kontinuerlig Glukose Overvåkningssystemer til å måle blodsukkernivået, og basert på denne målingen beregnes insulindosen og avgis via en insulinpumpe. Disse systemer er derimot ikke perfekte og lider av tidsforsinkelser fra starten på måltidet er oppdaget frem til insulin er administrert. Pasienter er påkrevd å kunngjøre måltids inntaket i tillegg til å kalibrere systemet på daglig basis. I et tidligere prosjekt ble arbeidet utført av Konstanze K. om automatisk deteksjon på starten av måltidet ved bruk av tarm lydopptak videreført, og svelge lydopptak ble innført. Dette prosjektet viderefører det arbeidet og tilpasser det slik at kun lydopptak av svelgelyder benyttes. I tillegg, ble en tale, måltidsstart, mattype, og svelge detektor bygget for å overvåke matinntaket, dette inkluderer varigheten på måltidet, mattype, og antall svelger. Dette studiet har visst potensialet av å bruke svelge lydopptak til å forbedre og unngå ulempene med en kunstig Bukspyttkjertel.

På første delen av prosjektet ble 10 svelge lydopptak brukt til å bygge en måltidsstart og en tale detektor. Ved å bruke disse opptakene ble effektspektraltetthet features beregnet, og brukt til å trene og bygge en Multilayer Perceptron klassifiseringsmodell. Tale detektoren hadde en nøyaktighet og en F1 score på 99%, mens måltidsstart detektoren hadde en nøyaktighet og en F1 score på 95%.

På den andre delen av prosjektet, ble 20 nye opptak tatt for å bygge en mattype og en svelge detektor. Disse opptakene ble hentet ved hjelp av to forhåndsvalgte mattyper, havre og salat. Opptakene ble samlet ved hjelp av to personer, hvor hvert opptak varte i ca. 30 min til 35 min, og mikrofonene som ble benyttet for å fange svelgelydene ble plassert rett over kragebeinet på halsen. Ved å bruke disse opptakene, ble Mel spektrogram features beregnet og brukt til å lage en Convolutional Neural Network klassifiseringsmodel. Svelge detektoren hadde en nøyaktighet på 93% og en F1 score på 92%, imens mattype detektoren hadde en nøyaktighet og en F1 score på 96%.

Preface

This thesis is submitted as a part of the Electronic Systems and Innovation master's project, in the subject TFE4940, in the Department of Information and Technology and Electrical Engineering. This project was provided by the Artificial Pancreas Trondheim (APT) group for the spring semester.

This project would not have been possible without the kind and meaningful support that the APT group and its people gave me throughout the semester. A special thanks must also go to my co-supervisor Salman Ijaz Siddiqui, who was with me in every step, guiding me and providing me with the help I needed. I would also like to thank my friends and family for their support and motivation and for always pushing me to continue to progress no matter the circumstances.

Contents

List of Figures

List of Tables

Nomenclature

CGM	Continues glucose monitoring
FN	False negative
FP	False positive
IMU	Inertial measurement unit
PSD	Power spectral density
TN	True negative
TP	True positive
ANN	Artificial neural network
ANOVA	Analysis of variance
AP	Artificial pancreas
BTS	Bolus transit sound
CNN	Convolutional neural network
DNN	Deep Neural Network
FDS	Final discrete sound
FP	False positive
IDS	Initial discrete sound
kNN	K-nearest neighbor
LSTM	Long short-term memory
ML	Machine learning
MLP	Multilayer perceptron
NN	Nerual network
ReLu	Rectified linear unit
RF	Random forest
RNN	Recurrent neural network

SGD	Stochastic gradient descent
SNR	Signal-to-noise ratio
SVM	Support vector machine

1 Introduction

1.1 Background

1.1.1 Diabetes

Diabetes is a chronic disease caused by the pancreas insufficient production of the hormone insulin. The hormone insulin is responsible for regulating the sugar level in the blood [10]. When food is consumed, it is broken down into glucose (sugar) and released into the bloodstream. Insulin acts then as a key to allow blood sugar into the body cells, in order to allow the cells to use energy [11].

The lack of sufficient insulin production can result in a high blood glucose level, also known as hyperglycemia. The longer the blood sugar level stays high the more damage it can cause the body. Hyperglycemia can lead to serious damage to the human body, especially to the nerves and blood vessels [10]. Hyperglycemia symptoms include weakness, shortness of breath, abdominal pain, and blurry vision[12] [7].

Diabetes patients are usually classified into two categories, type 1 diabetes and type 2 diabetes, pregnant women can also have a temporary form of diabetes known as gestational diabetes. Type 1 diabetes patients require insulin to be administered on a daily basis, as their body does not produce insulin on its own. Type 2 diabetes patient's bodies produce insulin, however, the body does not utilize it efficiently, thus the blood sugar is not kept at a normal level. Most diabetes patients have type 2 diabetes, around 90-95% of patients [11] [7]. Gestational diabetes is hyperglycemia that develops during pregnancy, the glucose level in the blood is higher than normal, however, it is less than those diagnosed with diabetes [13] [10].

1.1.2 Diabetes treatment

Prevention of type 2 diabetes is simpler than that of type 1 diabetes. A simple change to a healthier lifestyle could make a difference. Everything from eating healthier to being more active can help prevent or delay the onset of type 2 diabetes [11]. This is not the case for all type 2 diabetes patients, some patients are also required to take insulin for treatment [10].

Type 1 diabetes require insulin, therefore an insulin dose is given based on the blood glucose level. For this reason, a finger stick blood test is required before administering insulin. Insulin can be delivered today using a needle or a pen, or an insulin pump. The insulin pump is an automated solution that delivers insulin through a tube that goes under the skin. It is programmed to work as the body naturally would, by delivering insulin doses throughout the day [14]. Insulin pumps require a lot of information from the user throughout the day, including checking the glucose level by a finger stick blood test or manually increasing the insulin dose.

Other advanced diabetes solutions such as Artificial Pancreas (AP) systems rely on Continues Glucose Monitoring (CGM) devices. This type of system uses sensors that are inserted under the skin, into the fatty tissue [15], to measure the glucose level in the blood. Based on the sensor measurement, the needed insulin dose is administered. This type of system often requires calibration, thus the patient is required to measure the glucose level in the blood on a daily basis using a finger stick blood test [16]. CGM systems are subject to time delays of 30 min to 40 min from the meal onset until the meal is detected [17]. Hence, patients are required to announce their meal intake, this is a requirement by all clinically tested systems for glucose control [7].

1.2 Motivation

Today's insulin delivery solutions are not perfect, most treatments require the patient to be involved in every step of the therapy. This is not ideal, as the main idea behind these systems is to reduce the mental burden the treatment has on the patient, and also reduce the daily time spent on therapy.

To improve AP systems, an automatic meal onset detection system was proposed. The idea was to automate some of the steps in the treatment process, which in turn reduces the involvement of the patient during therapy. This system can remove the patient's need to announce the meal intake, in addition to reducing the meal onset detection time. Konstanze K. [17] tested the possibility of such an automatic meal onset detection system using bowel sound recordings. The study she conducted used a Support Vector Machine (SVM) classifier to detect the meal onset. Her study showed promising results, as the average meal onset detection time was reduced from 30 min to 40 min to approximately 10 min. Her system was however not perfect, as the recall and accuracy of the system were quite poor, and the proposed system suffered from a large number of False Positives (FPs).

In an earlier project, the work done by Konstanze K. was taken one step further by using a combination of both swallowing and bowel sound recordings as a means to detect the meal onset [7]. The proposed system used an SVM classifier. The addition of swallowing sound recordings helped reduce the detection time of meal onset to about 2 min, in addition to improving the overall recall and accuracy of the system. The proposed system worked well under a controlled environment, however, once the noise was introduced, the system performance was affected. When noise was introduced into the system in the form of speech, the amount of induced FPs increased, and the system struggled to separate between the meal onset and noise. In addition, the results from the previous project showed a weakness in using SVM as a classifier, because when the complexity of the problem is increased, the results are worsened.

Both studies showed that meal onset detection is possible using either swallowing or bowel sound recordings. Additionally, swallowing sound recordings could not only be helpful for injecting insulin at the right time, but could also provide relevant meal information, thus providing an estimate of the required insulin dose. This could help improve the current available AP solutions, and also reduce some of the burden diabetes treatment has on the patient.

1.3 Related work

Recent meal onset detection systems rely on different types of sensors. Some of the most commonly used types of sensors include inertial measurement unit sensors (IMU), microphone sensors, and piezoelectric sensors. This section is inspired by Muhammed U. and Huanhuan C. work on "Recent Trends in Food Intake Monitoring using Wearable Sensors" [18].

Piezoelectric sensors are used for automatic food monitoring by measuring force, acceleration, pressure, and strain. The way these sensors are used for detecting swallows is by utilizing vibration features.

Alshurafa et al. [19] used a wearable necklace that uses a piezoelectric sensor for swallowing and food type detection. The collected data is transmitted into a mobile app for processing. The acquired data are processed by calculating the standard deviation via a sliding window, before smoothing and filtering using a Savitzky-Golay convolution filter. Swallows are then found by counting the number of peaks detected after pre-processing, given a specific threshold for the spacing between the swallows. As for classifying the food type, swallowing spectrograms are divided into 16 bins and used to calculate statistical features such as skewness, mean, kurtosis, etc. A total of 360 features were calculated per bin. Three different classifiers were tested, k-Nearest Neighbour (kNN), Bayesian Network, and Random Forest (RF). The study showed that it was possible to distinguish between liquids and solids with an F1 score of 90%.

IMU sensors use force and angular rate for detecting the meal onset. These sensors capture body movement and provide relevant features that are used to obtain relevant meal information. A couple of IMU sensor-based approaches detected eating activity using wrist bands or smartwatches with built-in IMU sensors [20][21][22].

To detect meal onset Zhang et.al. [20] used a wrist band inertial sensor. Meal onset detection was achieved via a two-step approach, where in the first step the motion was detected, and the stationary period of individual movements was separated from other activities. In the second step, meal onset was detected via a motif-based approach, where motif candidates were searched for and then classified using an RF algorithm. This approach gave a 90% recall and an F1 score of 61%.

The research by Konstantinos K., Christos D., and Anastasios D. [21] relied on the inertial sensors of a smartwatch to detect eating events. By using the built-in six-axis inertial sensor in the watch, and an advanced motif-based time-point fusion technique, feeding gestures were detected. The classification system used Convolutional Neural Networks (CNN) for detecting certain movements, and Long Short-Term Memory (LSTM) for detecting eating sequences. Using this method, the recall was as high as 92% and the F1 score was as high as 88%.

In a recent study, Konstantinos K., Christos D., and Anastasios D. [22] did a similar study on detecting food intake cycles, by using the built-in inertial sensors in a watch. In this study, wrist movements and eating episodes were identified. Similar to their previous research, CNN was used to classify different movement patterns, before using an LSTM to classify the eating sequences. Using this method, a recall of 93% and an F1 score of 91% were achieved.

IMU-based approaches did not really have many swallowing-based meal onset detection methods, there were more chewing-based meal onset detection approaches [23][24]. Wang et al. [23] proposed a system that uses a single axis accelerometer to detect chewing activity, where frequency and time domain-based features are extracted and passed on to a classification algorithm that used Decision Trees (DTs), Multilayer Perceptron (MLP), SVM, Neural Networks (NNs) and a Weight-Supported SVM. This method provided a recall of 92% and an F1 score of 91%.

Christos M., Vasilis K., and N. Maglaveras [24] approached the detection of chews using 3-axis orientation data acquired by a smartwatch accelerometer. The features extracted from orientation are classified using an SVM. This method yielded 92% bite detection accuracy.

Microphone sensors had a large number of swallowing and chewing-based meal onset detection approaches [25][26][27][28]. The collected acoustic data could not only be used for meal onset detection, but also for medical applications in the field of dysphagia (difficulties of swallowing) and health/eating monitoring.

Oleksandr M., Paulo M., Stephanie S., Walter B., and Edward S. [25] used a microphone sensor to detect food intake episodes based on swallowing sounds. Their approach consisted of two steps, wherein the first step swallowing sounds were detected based on Mel-scale Fourier spectrum features. In the second step, an SVM classifier was used to identify swallowing sounds. Their system achieved accuracies of up to 80%.

Hajer K., Atta B., Dan I., and Jacques D. [26] work was about the detection of swallowing sounds, speech, and other ambient sounds through a neck-worn microphone for automated dysphagia patient care. The proposed system used a frequency-based analysis detection algorithm to distinguish between all three types of acoustic signals. The system also used an integrated automatic detection algorithm with the Gaussian Mixture Model (GMM) as a classifier. The classifier was trained using Mel-scaled features. The recognition accuracy of this system could reach up to 88.8%.

Temiloluwa O. Maysam G.[27] proposed a neck-worn system for the recognition of swallowing, coughing, chewing, breathing, speech, and heartbeat activity for health monitoring purposes. The system used a mix of time, frequency, and cepstral features to classify the different activities. The classifiers used for this system included kNN and Naive Bayes. Using these classifiers, the recognition rate for speech was in the range of 97.2% to 99.4%.

Yin B., Mingsong L., Chen S., Wenyao X, Nan G., and Wang Y. [28], used a wearable system to monitor and recognize food intake. The systems collected acoustic data were first pre-processed before being sent to a smartphone, where the food type is recognized. The chews and swallows were recognized first, using a Hidden Markov Model (HMM), before extracting time, frequency, and non-linear features. These features were then used on a DT-based algorithm to extract the type of food consumed. Using this method, the accuracy of food type recognition was about 84.9%, while differentiating between liquids and solid foods had an accuracy of 97.6% and 99.7% respectively.

1.4 Objective

As shown from the research conducted in the previous section, swallowing sound data could not only be used for automatic detection of meal onset, but also for classifying swallows and detecting the meal type. Hence in this thesis, a couple of detectors will be implemented using Artificial Neural Network (ANN) classifiers to monitor the food intake and to review the feasibility of using them for improving AP systems.

The main objective of this research will be to investigate the possibility of using swallowing sound recordings to improve the current treatment of diabetes patients. In order to do that, the goal will be to conduct a swallowing sound recording experiment as a means to help identify the meal type and detect swallows. For this purpose, a fixed protocol for administrating such an experiment is needed. Furthermore different types of features should be extracted and tested based on the studies in the section above to see what fits such a system. These features will then be used to build and test different ANN classification models.

Furthermore, as a continuation of the work done in a previous project [7], instead of using SVMs for meal onset detection, ANN models will be used. Speech detection will also be conducted to see whether it's possible to remove speech/noise from the meal region using an ANN classifier since SVMs did not work well in the previous project.

1.5 Contribution of this study

The main contribution of this study will be the four built ANN classifiers that could be used in future work to give an idea about how much food is consumed. The information obtained from these four detectors could also be used to improve some aspects of the current AP systems. This study showed also how different feature extraction methods affected the performance of the built classifiers, and how different problems encountered during the work on this project were tackled. Furthermore, the work done in this project included some relevant information and findings that could be relevant for future work in similar studies.

1.6 Thesis outline

This thesis is organized as such, first, the most important and necessary concepts needed for understanding the system and its different parts will be explained in detail in the theory section. After that, a short section about the equipment, the protocol, and the procedure used for the data acquisition will be presented. Following that, there will be two sections about the built systems. The first section will describe the method and implementation of the speech and meal onset detectors, in addition to presenting and discussing the results. The second section will describe the implementation method for the swallowing and meal type detectors, in addition to discussing and presenting the results of both systems.

To conclude this thesis, all the important findings in this thesis will be presented and summarized in the conclusion section. This includes the feasibility of using such a system to improve AP systems. Last but not least, there will be a short section that tackles future work.

2 Theory

2.1 Introduction

This section introduces the relevant theory needed to understand and support the research dealt with in this project. This section provides relevant concepts about swallowing sounds, signal processing, feature calculation, and machine learning.

2.2 Swallowing

Swallowing ensures that solid food or liquids are transferred from the mouth to the stomach via the pharynx and oesophagus [29]. The anatomy involved in the swallowing process is shown in Figure 1.

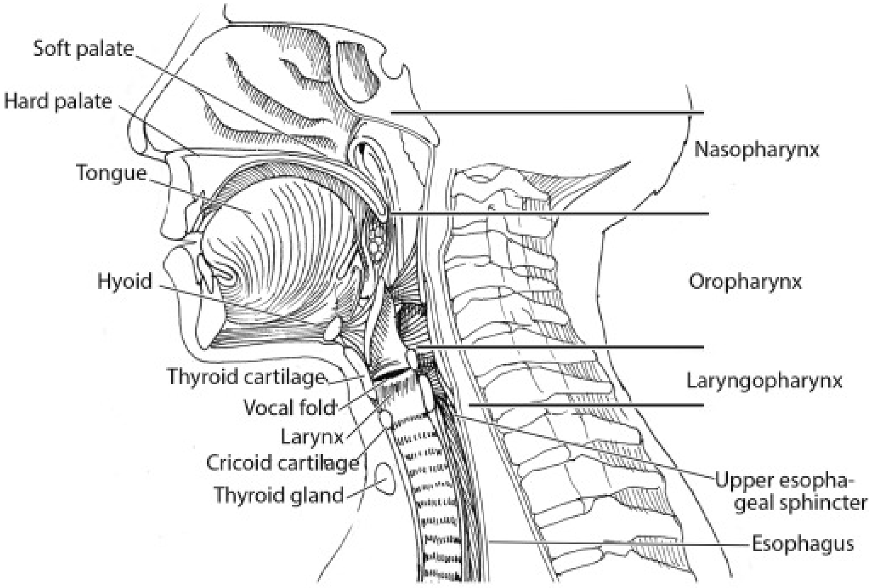


Figure 1: Anatomy of swallowing related organs [1].

Swallowing consists of four phases, which are the pre-oral phase, oral phase, pharyngeal phase, and oesophageal phase [30]. The pre-oral phase is the phase before the food enters the mouth when the person eating is anticipating the food, and the mouth starts salivating by the sight and smell of food. Then comes the oral phase, where the food enters the mouth and the chewing starts. The food is chewed into smaller pieces and mixed with saliva to form a bolus (a ball-shaped mass), before being transferred to the back of the mouth, at the pharynx.

The pharyngeal phase is where the reflex of swallowing is initiated. The soft palate is raised and the nasal cavity is sealed, which prevents any food or fluids from coming out of the nose. In this phase the larynx also moves upward and forward, thus the vocal folds close, and the epiglottis closes over the airway. In this step, the breathing stops for a brief moment. The bolus is then pushed down by the pharynx, by contraction. The oesophageal sphincter opens also up to allow the bolus to pass through. After passing the upper oesophageal sphincter closes, this prevents the bolus from moving back up.

In the final phase, the bolus is moved through the oesophagus via contraction until it reaches the stomach, where the lower oesophageal sphincter opens and closes to allow the bolus to enter the stomach, and prevent reflux.

2.2.1 Swallowing frequency range

Excluding the pre-oral phase of a swallow, swallowing sounds can also be divided into three phases, which are the Initial Discrete Sound (IDS), Bolus Transit Sound (BTS) and Final Discrete Sound (FDS) [26]. IDS is created due to the opening of the cricoid-pharynx in the pharyngeal phase, BTS is created in the pharyngeal phase when the bolus moves into the oesophagus. FDS is created in the final phase, the oesophageal phase. Swallowing different food types, liquids or solids generates different frequency characteristics at all the different stages.

The analysis made by Hajer Khalifi and co. [26] showed that swallowing foods with liquid-like consistency was associated with a frequency upper range of 3617 Hz. Swallowing water was associated with a frequency range of up to 2300 Hz, while dry swallowing (saliva) had a maximum frequency of 200 Hz. Other studies showed that the intensity of spectra in the frequency range 400 Hz to 1000 Hz significantly differed between liquids, semi-liquids, and solids [31]. For the purposes of this study, a frequency range of 0 Hz to 4 kHz was used in order to make sure that most meal information is kept.

In the frequency range below 1 kHz, noise sources such as heartbeat and breathing sounds could overlap with the frequency range of the swallowing sounds. Heartbeat sounds can have frequencies in the range of 20 Hz to 150 Hz [32], while breathing sounds can have frequencies in the range of 60 Hz to 700 Hz [33].

2.3 Signal processing

In this subsection of the project, most of the parts were reused from the previous project [7], as there were no substantial changes since no relevant new material was found during the work on this thesis.

2.3.1 Quantization

The number of bits of information per sample is the bit depth. When quantization is performed, the bit depth of the signal is reduced, this process leads to constraining the large set of values to a smaller one. Quantization leads to rounding of the values in the original signal, which in turn reduces the sharpness of the signal. This operation gives a lower signal-to-noise ratio (SNR), as a result of cutting some of the signal's highest peaks [34]. The frequency response of the signal is however unaffected by this operation since it's only constrained by the sampling rate of the signal.

2.3.2 Decimation

Downsampling is the operation where the sampling rate is reduced by keeping every M'th sample, where M is the downsampling factor. Lowpass filtering is not involved in the operation. Decimation is the operation where the sampling rate is reduced, but before that, lowpass filtering is applied to avoid aliasing. Aliasing is caused by downsampling with a sampling rate below the Nyquist rate. Decimation leads to a reduction in the power of the signal since the high-frequency content of the signal is attenuated [35].

2.3.3 Normalization

The goal of normalization is to use a common scale for the data without distorting the differences in the range of values. This helps with increasing the training speed of the classifier as the features used are in a similar range and values [36]. Linear normalization was used for this project, where the range of the values after normalization was between 0 and 1. Normalization is given by:

$$x' = \frac{x - x_{min}}{x_{max} - xmin} \tag{1}$$

Where x' is the normalized signal, x is the signal amplitude at a given time, and x_{min}, x_{max} is the minimum and maximum amplitude of the signal respectively.

2.4 Features

For this project frequency spectrum-based features were extracted to help train and test the built classification models. The features extracted in this project are represented in this subsection.

2.4.1 Power density spectrum

Power Spectral Density (PSD) features can be calculated using Fourier transform, and is given by [7]:

$$\hat{P}_s(f) = \frac{\Delta t}{N} | \sum_{n=0}^{N-1} x_{s,n} e^{-i2\pi fn} |^2 \tag{2}$$

Where Δt is the sampling interval and N is the number of samples in time segment s [37]. This feature shows how the signal energy is distributed in the frequency domain. For this project power in different frequency bands in addition to the power fraction in these bands was calculated.

2.4.2 Mel spectrogram

Mel frequency features are widely used for speech recognition. As the name suggests, Mel scale is used, which allows mimicking of the hearing perception of humans for machines, as humans tend to have higher resolution for lower frequencies [38]. Mel scale transforms the frequency scale into a non-linear frequency scale.

Mel spectrogram is just like a normal spectrogram that shows the power of the signal over time and frequency, however, the frequency is plotted using the Mel scale. Mel spectrogram is calculated by first breaking down the audio signal into frames with overlap, and then for each frame, the Fourier transform is calculated. Then the signal is converted into a Mel scale using a triangulation bandpass filter. These filters are wider at higher frequencies to mimic human hearing. For each frame, the magnitude of the signal is decomposed into components that correspond to the frequencies on the Mel scale [39]. In this project, the signal was decomposed into a total of 128 frequency components, meaning there was a total of 128 features.

2.4.3 Mel frequency cepstral coefficients, Delta and Delta-Delta

The steps for calculating the Mel frequency cepstral coefficients are illustrated in Figure 2.

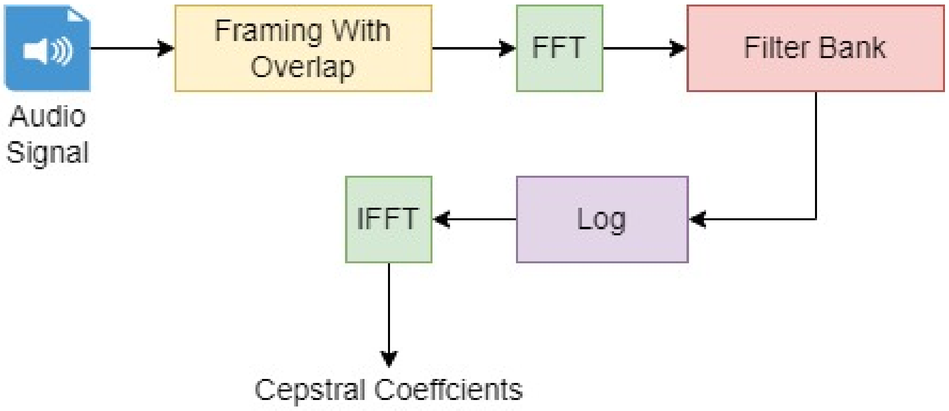


Figure 2: Block diagram of the steps to calculate Mel frequency cepstral coefficients.

In the first step, the audio signal is broken down into frames with overlap. It is common to use a hamming window to frame the signal, this keeps the original frequency information while maintaining less noise. After framing, the Fourier transform is applied to each frame in order to extract the frequency information. After this step, frequency is mapped into the Mel scale using triangular bandpass filters, before taking the logarithm. The logarithm makes it possible to mimic the power perception of humans, as humans are more sensitive to small changes at lower power than at higher power. In the last step, the inverse Fourier transform is applied. Most of the information is kept within the first few coefficients [40] [41]. Hence only the first 13 coefficients are often used.

It's common to calculate Delta and Delta-Delta features alongside MFCC features. These features represent the evolution of the MFCC over time. These features are great for representing time-varying signals such as speech, and in this case, swallowing. Delta coefficients are computed using Equation (3), where d_t is the Delta coefficient at frame t computed from the MFCC coefficients c_{t+n} and c_{t-n} at frame $t \pm n$, n is usually set to two [42].

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{3}$$

Delta-Delta coefficients are also calculated using Equation (3), but in this case the Delta-Delta coefficients are computed using the Delta coefficients c_{t+n} and c_{t-n} [40]. Both MFCC and MFCC-delta's features were tested early on in the project but were later on dropped, as they did not contribute to the final built models.

2.5 Machine learning

Machine learning (ML) enables the system to learn and improve through experience with the help of data. This enables the system to solve a multitude of problems in different fields from business and art to science and technology. ML algorithms can be classified into three different categories, which are supervised learning, unsupervised learning, and reinforcement learning.

In supervised learning, the model is trained using labeled data to solve classification and regression problems. In unsupervised learning, the model is trained using unlabeled data. Unsupervised problems include clustering and data analysis. Reinforcement learning is different from the other two methods as the model is trained by trial and error, where the main goal of the system is to maximize the rewards [43].

This project tackles a supervised learning problem, a classification problem, where the algorithm is trained to assign the data to a specific class. The main classifier used for this purpose is an Artificial Neural Network.

2.5.1 Artificial neural networks

Artificial Neural Network (ANN) are learning algorithms that are inspired by the way biological Neural Networks work. ANNs are great at recognizing patterns in any type of data, either its images, text, sound, etc [44].

Each ANN consists of a large number of nodes, where calculations take place. The nodes from each layer are interconnected, and each connection is weighted. In each node, the weighted node's connections are summed. These weights decide the significance each node has on the next layer, also each node has a bias connected to it, which offsets the node's output. The summed inputs are passed through an activation function that decides whether the output of the node is activated or not, depending on a pre-defined threshold [45]. A typical node can look like this, Figure 3, where a_1 to a_N are the inputs from the previous nodes, and a_{out} is the output from the current node.

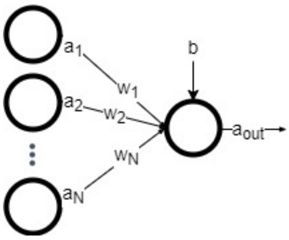


Figure 3: Typical inputs and output of a node, where a_1 to a_N and w_1 to w_N are the previous node's outputs with the corresponding weights, b is the bias, and a_{out} is the current node output.

A typical ANN consists of three types of layers, an input layer, a hidden layer, and an output layer. Each layer consists of multiple nodes. The input layer is the first layer of the network, and usually has the same number of nodes as features or inputs. Following the input layer is the hidden layer, most of the computation is carried out here [46]. Lastly is the output layer, this layer is responsible for producing the results, and has the same number of nodes as classes or required outputs. A typical ANN is shown in Figure 4, which is also known as a Deep Neural Network (DNN), as it consists of multiple hidden layers (more than three).

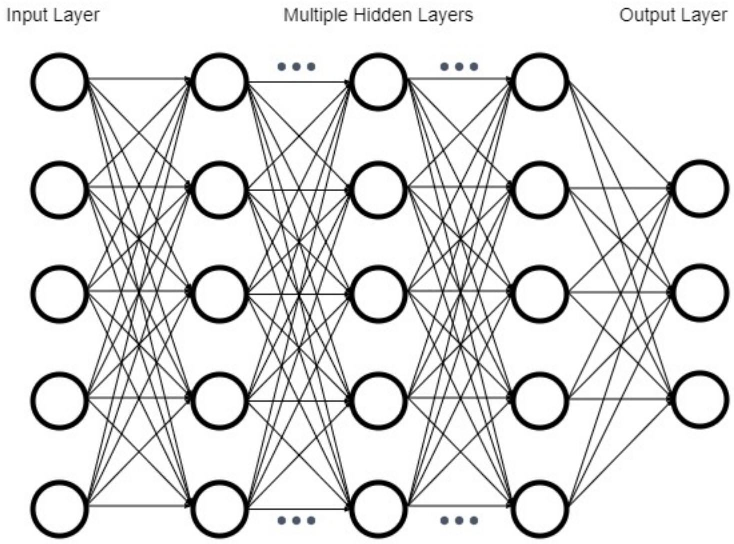


Figure 4: Typical Deep Neural Network structure. It consists of more than three hidden layers.

2.5.2 Activation function

Activation functions simply decide whether a node is activated or not (fires or not), given certain criteria are met. These functions are only used in the hidden and output layers. The output of the neuron (Y) is given by the sum of the weighted inputs from other neurons, and the bias of the neuron, (4) [47].

$$Y = \sum(Weight \cdot input) + bias \tag{4}$$

The output of the node can, in theory, have any value, this is where activation functions come into play. The activation function checks the Y value in order to decide whether the neuron in the next layer will use the output from the current node or not. The activation function also reduces the range of outputs.

There are two main types of activation functions, linear and nonlinear. For the task at hand, a non-linear activation function was used, as the problem itself is non-linear. The nonlinear activation function of choice for this project was Rectified Linear Unit (ReLu). This function outputs zero if the input is negative and returns the same value for positive inputs. The function could be written as shown in (5) [2], where max is a function that returns the input value as long as it is equal to or greater than zero.

$$f(x) = max(0, x) \tag{5}$$

Graphically ReLu can look like the plot shown in Figure 5. Even though the function looks simple, it works really well for most classification problems, ReLu is in fact one of the most popular activation functions [48].

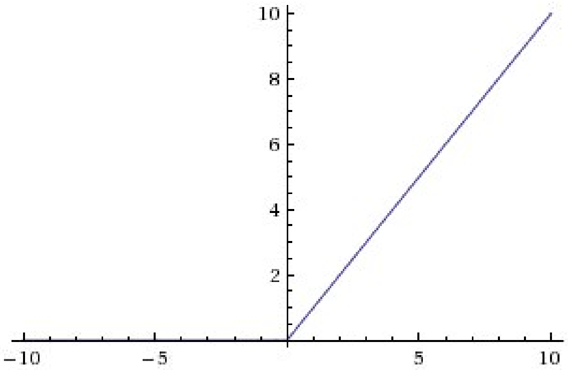


Figure 5: Graph of the Rectified Linear Unit Function [2].

The activation function serves two primary purposes, the first is to help the model account for interaction effects, and the second is to help the model account for non-linear effects [2]. Interaction effects are in essence how some inputs of a given node might dominate the output by having a large value, making the output either zero or positive. These interaction effects increase even more with the increase in the number of layers and nodes. As for capturing non-linearity, ReLu is non-linear around zero, as it changes from zero to positive, thus it has limited non-linearity [2]. However, due to the biases in each node, the slope moves where it changes, and due to the fact that there are many nodes, this allows for many slope changes, which in turn produces nonlinearity.

The output layer usually uses a different activation function than that of the hidden layers, and this function is selected with regard to the desired output of the ANN. The problems in this project were either binary or multi-class classification problems. For the purpose of binary classification, Sigmoid is usually used as the activation function for the output layer. This function converts the input into probabilities between 0 and 1, and the output is interpreted as one class or the other depending on the probability value, whether it is larger or smaller than 0.5 [49]. The function is shown in (6), where z is the input of the node.

$$Sigmoid(z) = \frac{1}{1 + e^{-z}} \tag{6}$$

For multi-class classification problems, softmax is used as the output activation function. Each input is converted into a probability value between 0 and 1, where the sum of all the outputs is always 1. The function is shown below, (7), where the node's input z_i, is normalized over all the output layer outputs, z_j [50]. The sigmoid function can be seen as a special case of a softmax function, where one of the two output nodes has no weights.

$$softmax(z_i) = \frac{exp(z_i)}{\sum_j exp(z_j)} \tag{7}$$

2.5.3 Training, validation, and test set

To build a machine learning system, it is common to have three types of data, training, validation, and test data. Training data is used only to train the model and update the parameters. Validation data is only used to tune the hyperparameters and check for overfitting. The test set is used to test the final model, and check how well it performs on unseen data, this data must not be used at any stage during the training of the model [7]. The validation set is often confused with the test set, however, they are not the same.

2.5.4 Training the network

This subsection is inspired by Tor Andre M. signal processing course, TTT4135 [51]. The parameters (weight and bias) in the model are usually initialized at random and then updated during training. In order to train the model, there should be a measure that describes how good the output of the model is, a measure of error, and hence a loss function is defined. The most commonly used loss function for classification problems is the cross-entropy loss, which is given by (8), where y is the true label and $\hat{y}$ is the predicted label.

$$l(y, \hat{y}) = \sum_i -y_i log(\hat{y}_i) \tag{8}$$

Given a loss function, the total loss for the given training problem can be defined as:

$$L(\Theta, X, Y) = \frac{1}{|X|} \sum il(y_n, f(x_n; \Theta)) \tag{9}$$

Where Θ is the parameters of the network, while X and Y are the training data and their labels. The goal of training the neural network is to minimize the total loss with respect to the parameters as shown in (10). However, this problem cannot be solved in a closed form, and must therefore be solved by numerical search, usually using the gradient descent algorithm.

$$\hat{\Theta}, = \underset{\Theta}{\mathrm{argmin}} L(\Theta, X, Y) \tag{10}$$

The gradient descent algorithm optimizes the network and reduces the loss, however, to achieve this, the gradient must be calculated. The back-propagation algorithm is used to calculate the gradients, and it does so by first looping through the network and doing the forward computation where the intermediate results are stored. The stored results are then used to calculate and update the gradients. This works because it can be shown that the gradient Θ^L depends on the next gradient Θ^{L+1}.

When looping through the data, one of the following three approaches could be used, Batch, Mini-Batch, and Stochastic Gradient Descent (SGD). In Batch all the training data is used to compute and update the gradients, this leads to slow convergence per epoch (complete pass through the training set). In Mini-patch only a small subset of data is used for computing the gradient, thus the gradients are less accurate, however, the model is updated multiple times per epoch and converges faster. Lastly SGD uses only one training sample that is selected at random to compute the gradient. The gradients are even less accurate than for Batch, however, the gradients converge faster than that of Batch.

For this project, an extension of the gradient descent algorithm was used, which is called the Adaptive Movement Estimation algorithm, or Adam in short. The algorithm "adapts a learning rate for each variable for the objective function and further smooths the search process by using an exponentially decreasing moving average of the gradient to make updates to variables" [52].

One of the essential hyperparameters for learning the network is the learning rate for the gradient descent algorithm. As the name suggests it's responsible for how fast the network learns, and thus how fast the parameters are updated. The selection of this hyperparameter depends on many factors, such as the topology of the network and the data set. This hyperparameter is usually tuned throughout the building process of the model, in order to make sure that the loss is decreasing per epoch [53]. The effect the learning rate has on the loss is illustrated in Figure 6, as can be seen, the value must not be too high or too low, as both can cause a high loss.

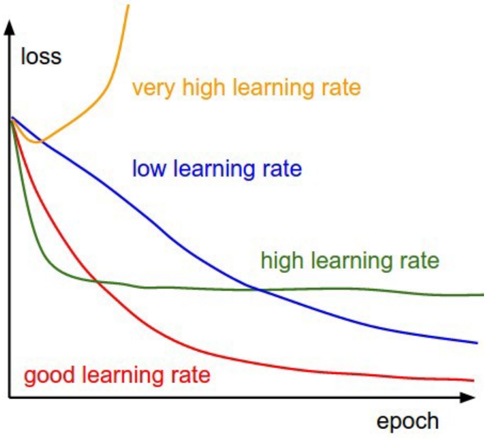


Figure 6: Illustration of the model's loss over each training epoch, given different learning rates [3].

2.5.5 Overtraining

Overtraining becomes a problem when the model starts learning the training data too well, this makes the model's predictive ability bad. This problem can occur due to multiple factors, such as when the model is too complex for the problem at hand, or when there is little training data. Overtraining is usually detected by plotting the loss against the number of iterations for both the training and validation set, as illustrated in Figure 7. The model is overtrained if the training set loss keeps decreasing while the validation set loss starts increasing. When this occurs the model's generalizability on new data becomes horrible, this leads to a low training loss and a high test loss.

Another problem that can occur when training the model is undertraining, which is the exact opposite of overtraining. This happens when the model has not been trained long enough. This leads to a high training loss and a high test loss. The best scenario is to find the sweet spot, where the model is balanced, having low training and test loss [54].

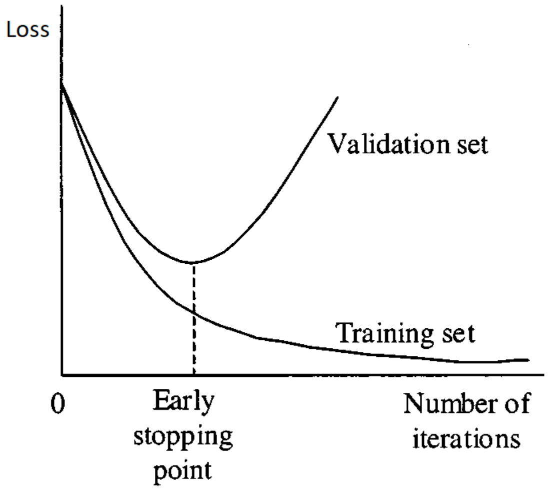


Figure 7: Illustration of the loss curve for both the training and validation set, where the early stopping is marked by a dotted line [4].

To avoid overtraining methods such as early stopping, feature selection, regularization, and dropout could be used. Early stopping is about stopping the model's training before it starts overtraining by reducing the number of epochs, just as illustrated in Figure 7. Regularization is the process where parameters with large coefficients are penalized using a penalty term. The parameters with large coefficients are usually the ones responsible for limiting the variance, which then leads to overfitting [43]. Dropout helps with overtraining by discarding a number of node connections so that they do not contribute to the model's output, and those connections are chosen at random. This helps to break down dependencies in the model.

Feature selection helps with reducing the complexity of the model, as the redundant and nonrelevant variables are removed, this also helps with overtraining. For this project Analysis of Variance (ANOVA) F-test was used for feature selection. The test calculates the relationship between two variance values, in this case, the input variable and the target variable, and determines whether it comes from the same distribution or not [55]. The more likely that it comes from the same distribution, the higher the score the variable gets, this makes it more likely to be selected, and vice versa.

2.5.6 Types of Neural Networks

There are many types of ANN out there, some of the most popular ones are Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). For this project only MLP and CNN were used, both are described below. RNN was not explained, as it's outside the scope of this study.

Multilayer perceptron

MLP is composed of a series of fully connected layers and is the most basic ANN out of the other two. This network is a class of Feed-Forward ANN [56] and thus has the same structure as the DNN shown in Figure 4.

Convolutional neural network

Unlike MLPs, CNN can take images as inputs. CNN uses convolution layers that carry convolution operations, this allows the network to capture temporal and spatial dependencies. Each convolution layer has a number of filters/kernels that compute features from the images. The features extracted include lines, edges, shapes, etc.

In addition to convolution layers, CNN uses also pooling layers that reduce the dimensionality of the input by the use of statistics such as max, mean or average. For this project Max-pooling was used, this returns the maximum value from the image portion that is covered by the kernel.

CNN includes also Nonlinear processing by having a fully-connected layer at the end of the model, an MLP. This usually comes after flattening the output of the images, so that it's of suitable form for the MLP [57]. A typical CNN for a multi-class classification problem is shown in Figure 8, where the input image is classified into one of three classes, based on the class output with the highest probability.

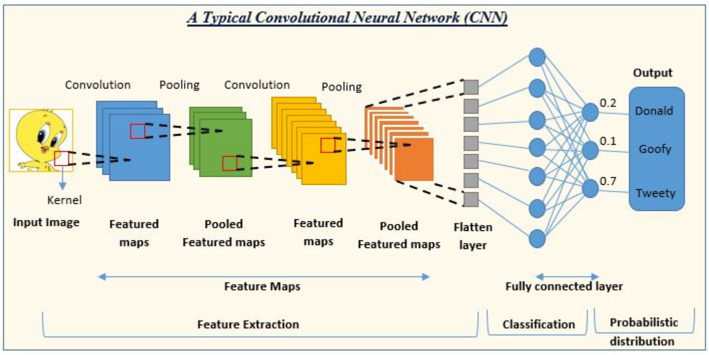


Figure 8: A typical Convolution Neural Network, and an example of a multi-class classification problem [5].

2.5.7 Performance assessment

To assess the performance of the built ANN models, learning curves were used. Learning curves are a great way to see whether the model is being overtrained or undertrained. The loss and accuracy learning curves are usually plotted for both the training and validation set.

Some common classifications metrics such as True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN), are also used in this project. TP describes how many correctly classified predictions there are for a given class. FP describes how many falsely classified predictions there are for a given class. TN is when you predict that an observation does not belong to a class correctly. FN occurs when you predict that an observation does not belong to a class falsely [58]. All of these are usually summarized in a confusion matrix, an example of a binary confusion matrix is shown in Figure 9.

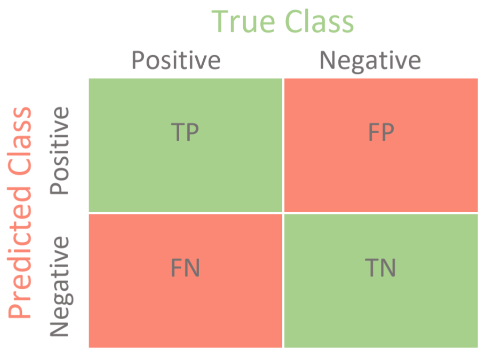


Figure 9: Binary confusion matrix [6].

Other metrics such as accuracy, precision, and recall are also commonly used to describe the performance of the system, these matrices are computed using the confusion matrix. Accuracy is defined as (11), where Total is the number of all classified elements. Accuracy is a measure of how often the classifier predicts correctly. Precision is defined as (12), and is a measure of the quality of the classifier. Last but not least, Recall is defined as (13), and describes the quantity, that is, how many of the relevant items are selected [58].

$$Accuracy = \frac{TP + TN}{Total} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

Recall and precision can be combined into a single metric, also called the F1 score. This is defined as shown below:

$$F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{14}$$

All these matrices and plots are of importance, especially in the early phases of building the ML models. They were actively used during this project, to improve the results and build the best possible classification models.

2.6 Python

Python is a great programming language, especially for machine learning applications. It has a lot of great libraries for different applications. Libraries such as TensorFlow, and Librosa have many great built-in functions for calculating features and building ANN models. It has also a large number of great tools to analyze and deal with data. For this project, Google Colabroratory was used for programming in Python, as it allowed for free GPU access on the Web.

3 Equipment and protocol for data acquisition

3.1 Introduction

This section presents the equipment and protocol used in this project. Some of its parts are inspired by the previous project [7], since the same equipment has been used. The protocol used in this project was inspired by Konstanze K. protocol in the pilot study "Analysis of bowel sounds related to meal onset" [59], in addition to the paper on "Automatic food intake detection based on swallowing sounds" [25].

3.2 Equipment used

The equipment used for this project is shown in Figure 10. The microphones used for recording are connected to the red box, which is the power supply of the microphones. There were a total of four SPM0687LR5H-1 microphones, each one of the four microphones was placed inside a stethoscope/disc-shaped holder, this design was intentionally like this to make capturing sounds made by the human body easier. These microphones are omnidirectional and have a uniform frequency response and a high SNR [60].

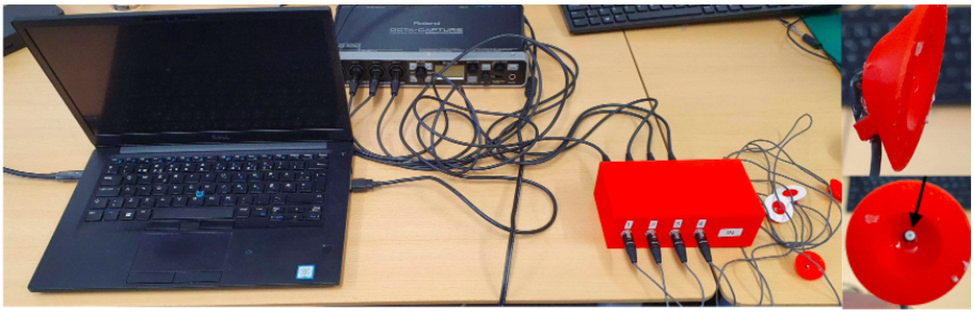


Figure 10: Equipment used for data collection, The microphone is shown on the right lower side of the image, pointed to by the black arrow.

The output of the microphone is connected to a sound card, the Roland Octa-Capture sound card. The sound card takes in the analog sound data from the microphone and digitizes it. This sound card has a resolution of 24 bits and a sampling frequency of 48 kHz [61]. All the recordings are initiated and saved in the laptop connected to the sound card, before being stored on the cloud, on Google drive. A diagram of the equipment used for recording is shown in Figure 11.

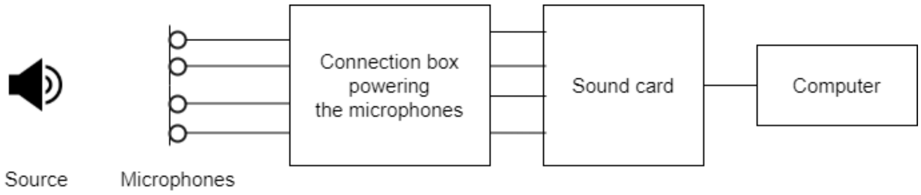


Figure 11: Representation of the equipment layout during the data collection [7].

3.3 Protocol for meal type recordings

For the meal type experiments, it was necessary to have two meals with a distinct difference in consistency, for this purpose salad and oats were selected. For the oatmeal, a pre-packaged oat in a cup of 54 g was used, while for the salad meal, a pre-packaged salad plate of 225 g was used, as shown in Figure 12. The salad meal contained also pasta, chicken, and dressing, only the pasta, salad, and dressing were consumed in each meal, this accounted for roughly 78% of the weight of the meal. Thus, the salad meal was actually around 175.5 g. The salad, dressing, and pasta were mixed before each salad meal recording, in order to ensure that every bite contained salad. The oatmeal had to be prepared using hot water, it was prepared exactly as instructed on the package before each oatmeal recording. Pre-packaged meals were used to make sure that the same amount and same meals were consumed in each recording.

Figure 12: The prepackaged oat and salad meals were used for the meal type recordings. The pictures are brought from the store’s website, "Meny" [8][9].

The meal type experiment recordings followed the proposed protocol shown in Figure 13. The first 20 min of the recording were used as a reference, there was no talking or eating during that period. After that, the subject was able to eat either of the proposed meals, either oats or salad. The meal duration could not exceed 10 min, which was manageable as the meal portions were quite small. After the meal ended, the recording continued for another 5 min, again there was no talking or eating during that period. In total the whole recording lasted between 30 min to 35 min.

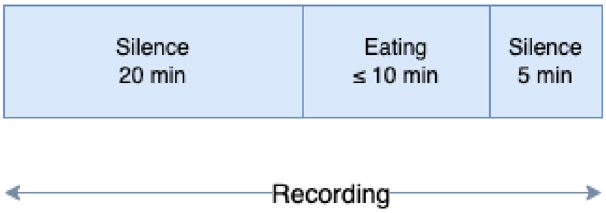


Figure 13: Protocol for meal type recordings.

3.4 Old recordings protocol

Some recordings from the previous project [7] were also included for use in this project, mainly for the speech and meal onset detection part of the project. These recordings had a similar protocol to the meal type recordings, however, some parts were longer, and some parts included noise in the form of speech. There were two types of recordings acquired in the previous project, some recordings were augmented with noise, while the other recordings had no noise, only the meal onset [7].

The recordings without noise had a protocol like the one shown in Figure 14. The subjects were asked to fast for a minimum of 3 hours before the recordings, as, unlike this project, bowel sound recordings were used in the previous project. The recordings started by first using the first 15 min as a reference, there was no meal or talking during this period. After that the subject could have a meal of their choice, usually, the meal consisted of a slice of bread with some cheese or other toppings and a glass of water. After that the recording continued for another 45 min, once again there was no talking or eating during that period, this was to monitor the digestive behavior.

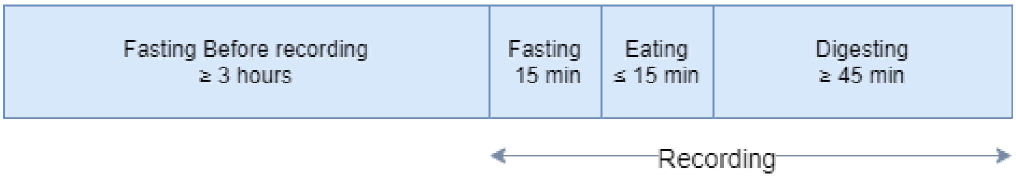


Figure 14: Protocol for recording data without noise [7].

The noise-augmented recordings had a protocol similar to that shown in Figure

15. These recordings were intended to represent a more realistic environment. This protocol was more focused on the swallowing recordings, just like the meal type protocol, thus fasting before the recording was not needed. The recordings consisted of 5 min of silence, followed by 10 min of reading from a book or news article. The first 15 min were used as a reference. Then the subject could eat once again a meal of their choice, for up to 15 min. Then after the meal, there was 5 min of silence followed by 5 min of reading. In addition to that, two of the meal recordings augmented with noise had also speech during the meal onset, the subjects were asked to eat normally, then speak for 10-15 s, and then continue eating, this was repeated throughout the meal onset period. This was mainly to see how the model would react to speech in the meal onset region.

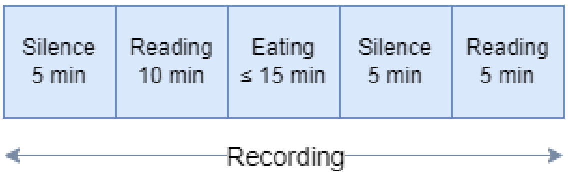


Figure 15: Protocol for recording data augmented with noise [7].

3.5 Recording procedure

This recording procedure yielded all the recordings, both the old recordings and meal type recordings. For each recording all four microphones were used simultaneously, the placement of the microphones is shown in Figure 16. "The swallowing microphone was placed just above the collar bone on the neck to record swallowing sounds. Different microphone placements were tested in the paper "Automatic detection and recognition of swallowing sounds" [26]. Based on the findings in the paper, the best placement appeared to be just above the collar bone, since these recordings had the highest power. This location was, however, not the most comfortable with regard to head movements." [7].

Two of the other three microphones were placed on the lower abdominal region, these were used to capture bowel sounds. The third microphone was placed 2 cm below the right ear to capture chewing sounds. The recordings captured from these three microphones were not used in this project, they were only captured for use in other projects.

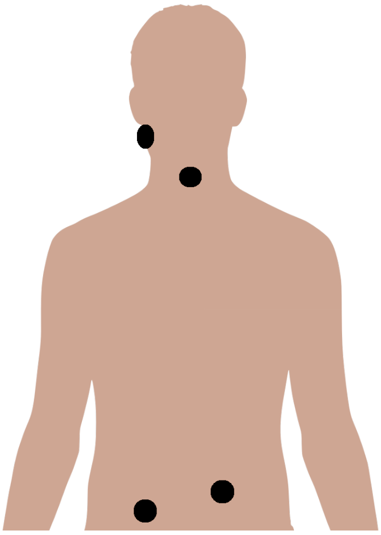


Figure 16: Locations of the four microphones [7].

The microphones were fastened on the specific locations using a double-sided tape, that was placed in the outer ring of the microphone, as could be seen for two of the microphones in the lower right part of Figure 10. This ensured that the microphone stayed in place during the entire recording process. All the recordings were obtained in a quiet room, with as little noise as possible. For all the recordings, the subjects were seated on a reclined chair and asked to move as little as possible, in order to minimize any unnecessary noise that could affect the quality of the recordings. Subjects could read a book or use their phone throughout the experiment period to avoid boredom, as long as they did not move too much.

4 Speech and meal onset detection

This section introduces the built meal onset and speech detection systems, these were built using an MLP classification model. These two systems were built to improve the results from the previous project [7] and to test whether an ANN classifier performed better than an SVM classifier for such a problem.

4.1 Method description and implementation

This section describes how the speech and meal onset detection systems were implemented. All the steps required for building the systems, including pre-processing, feature extraction, training, and classification, will be explained in detail.

4.1.1 Data acquisition

For this part of the project, the data collected in a previous project were used [7]. There were a total of 10 recordings, 5 of them were augmented with noise, while the other 5 were not augmented with noise. All recordings were acquired using the protocols and procedures described in sections 3.4 and 3.5. These recordings were collected by 2 subjects, this ensured some variability, even though both subjects were around the same age (mid-twenties), and had no history of health problems related to swallowing.

These recordings were meant to test the speech detector's ability to classify speech in the recordings, and whether the meal onset region in the recordings affected the classification or not. As for the meal onset detector, the system's ability to distinguish between the meal onset and noise was tested, this included friction noise from the movement of the microphones, talking, and other noise sources. In order to make use of all the available data, all of the recordings were used for both systems, since during this stage of the project, there were no other accessible recordings data.

4.1.2 Data pre-processing

The recordings obtained using the "Roland Octa-Capture" sound card had a bit rate of 24 bits, the bit rate was reduced to 16 bits by quantization. This was to reduce the size of the recordings before working with them on the cloud, as they were too large to handle on Google drive because most of the data were stored there. The quantization was performed using Audacity, which is an audio editor. Audacity was used only for simplicity reasons. This process was also convenient since most of Python's built-in libraries supported 8-bit, and 16-bit data out of the box.

After quantizing the recording data, the data was decimated. This was a two-step process, where first the data was filtered and then downsampled. Decimation was applied using either a bandpass or a lowpass filter. A lowpass 5'th order Butterworth filter with a cutoff frequency of 4 kHz or a bandpass 5'th order Butterworth filter with a cutoff frequency of 80 Hz and 4 kHz were used. Both filters had a linear phase response in the region of interest, in addition to a flat amplitude response. This helped with keeping the original signal shape after filtering. In addition, at the cut-off frequency, the magnitude had a -3 dB amplitude drop for both filters. The frequency and phase response of the filters could be seen in Figure 17, as could be seen the graphs for both filters were overlapping. This was the first of many other variations to test how the performance of the system was affected by such variations.

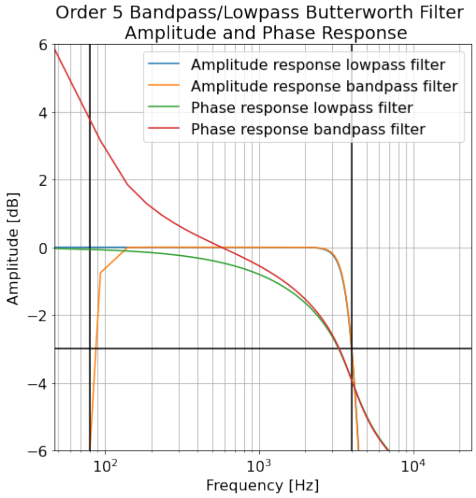


Figure 17: Amplitude and phase response plot for the decimation filters. The black lines illustrate the cut-off frequency and magnitude drop by -3 dB. Amplitude responses for both filters are overlapping.

The cutoff frequency for the lowpass filter was selected based on the literature, and also the spectrogram of the raw recordings which is shown in Figure 18. The spectrogram is from one of the recordings augmented with noise, it was plotted for a duration of 2 min. In the first minute, the subject was talking, while in the second minute the subject was eating. It was clear that most of the power for the meal onset was concentrated for the frequencies below 1.5 kHz. Speech could easily reach frequencies up to 20 kHz. Since meal information is more important than speech for this project, the selection of the low cutoff frequency of 4 kHz would not affect the performance of the system. Also, some high-power speech information is kept for frequencies below 4 kHz.

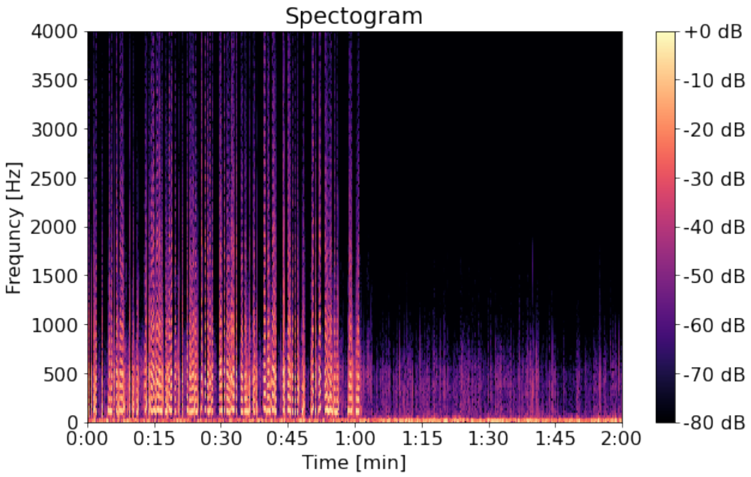


Figure 18: Spectrogram of raw meal recording, where the first minute contains talking, while the second minute contains only meal intake.

The cutoff frequency for the bandpass filter of 4 kHz was selected for the same reasons as mentioned above, while the cutoff of 80 Hz was selected to reduce some of the noise from heartbeat and breathing sounds while keeping most of the relevant meal information intact. After filtering the signal was downsampled to 8 kHz, this reduced the total amount of data by a factor equal to the downsampling factor, $L = 6$.

4.1.3 Feature extraction

For both classification systems, frequency spectrum-based features were extracted. PSD features were extracted, since, in the meal and speech region of the recording, the power is much higher for the frequency range of interest, as when compared to non-speech and non-meal regions in the recording. In addition, the power level is also different for speech and the meal onset regions of the recording, speech tends to have more power. In addition, PSD features proved to work well for meal onset detection using an SVM classifier in the previous project [7].

Before calculating PSD features each recording data was first normalized individually, by simply dividing by the max value within the data. PSD features were then calculated by first segmenting the data into segments of equal length with 50% overlap. Three segmentation lengths were tested, 5 s, 10 s, and 15 s. This was done to see whether segmentation length had an effect on the performance of the classification systems or not. For each segment, power features were calculated for each 100 Hz band in the frequency range from 0 Hz to 4 kHz, using the formula in Equation (2). In addition to that, the total power in the frequency range from 0 Hz to 4 kHz, and the power fraction for each 100 Hz band in the frequency range from 0 Hz to 4 kHz were also extracted. These features are summarized in Table 1.

Features	Number of features
Total power	1
Power in 100 Hz frequency bands from 0 Hz to 4 kHz	40
Power fraction in 100 Hz frequency bands from 0 Hz to 4 kHz	40

Table 1: Description of the extracted Power Spectrum Density features [7].

After calculating the PSD features, each feature was then normalized using (1), which scaled the features to values between 0 and 1. The normalized features were then used to build a feature matrix for all the recording data. Where the columns represented the features, and the rows represent the time segments, as illustrated in Figure 19.

$$\begin{vmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,81} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,81} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,81} \end{vmatrix}$$

Figure 19: Feature matrix, where columns correspond to features at a given time segment, and n is the segments in the recording data.

4.1.4 Data splitting

After feature extraction, the data was split into training, validation, and test set. Since there were initially only ten recordings, all of the recordings were used for training and testing both models. Three recordings were assigned for testing the model, two of which were augmented with noise. As for the training and validation set, a total of seven recordings were used, six for training and one for validation.

The test set recordings were selected beforehand because two of the recordings augmented with noise had also speech during the meal onset. This was done to see how speech in the meal onset duration is classified since this affected the built SVM classifier in the previous project [7]. The built SVM classifier struggled to separate speech from the meal onset. The last test recording was selected out of the recordings without speech, the recording was selected at random. The rest of the recordings were assigned to the training and validation set at random.

4.1.5 Training and classification

Before training and building an ANN model, a labeling vector is needed. This vector is in essence the target vector that the ANN model will try to predict when training the model. For the speech detection model, a label of "1" was given for each segment that contained speech, while all the other segments were given the label "0". Similarly, for the meal onset detection model, a label of "1" was given for each segment that contained a meal onset, while the non-meal segments were given a "0". Since there are two classes in each model, meal or no meal and speech or no speech, both problems are considered to be a part of a binary classification problem.

After building the labeling vector, the feature selection algorithm was used, in this project, ANOVA F-test was used. Different feature selection percentages were tested while building the model, as the performance of the model varied depending on the number of selected features. This percentage was fixed when the final classification model was built. For the speech detector, 35% of the features were selected, 28 features in total. As for the meal onset detector, 50% of the features were selected, a total of 40 features. This step is important to reduce the complexity of the classifiers as much as possible, help with overtraining, and also reduce the training time.

When building an ANN model, there is not really a specific method, different things must be tested in order to obtain a working model. Hence when building the model, a different number of layers, nodes, learning rates, epochs, activation functions, and batch sizes were used. Often the first thing that is tested is the number of nodes in each layer, just to have a grasp on what works, whether it's a large or a small number of nodes. After that, a different number of hidden layers is tested, including different activation functions, until the results are stable. Then the learning rate, number of epochs, and batch size are varied until the results stop improving. When the results are good, some of the methods to avoid overtraining are used. The model that gave the best results with regard to the test set is usually selected. For starters, this building process was first executed using the bandpass filtered 10 s segment features, these features were used as a reference for both the meal and speech detectors.

When a functioning model is built, and the results stop improving, the architecture of the model is set. When that is the case, then the model is tested using both the lowpass and bandpass filtered features for the 10 s segment. The results are compared, and the features that give the best results are used in the next comparison step. In the next comparison step the features using the best filter and segmentation lengths of 5 s, 10 s, and 20 s are compared. When the most suitable feature extraction method is selected, the model is then tuned one last time to ensure that the best possible results are achieved before selecting the final architecture.

The final system for the speech detector is shown in Figure 20. The final architecture used for this detector was that of an MLP. The model consisted of three layers in total. The input layer was a flattening layer, with 28 Nodes, one for each feature. This was used to convert the data into a 1D shape to allow for calculations in the next layer. Then there was a hidden layer consisting of 200 nodes, with a dropout rate of 30%, the activation function used for this layer was ReLu. Most computations take place in this layer. Lastly, the output layer had only 2 nodes, one for each class, and since this is a binary classification problem, Sigmoid is usually used as an activation function, however, softmax was used as it provided better results.

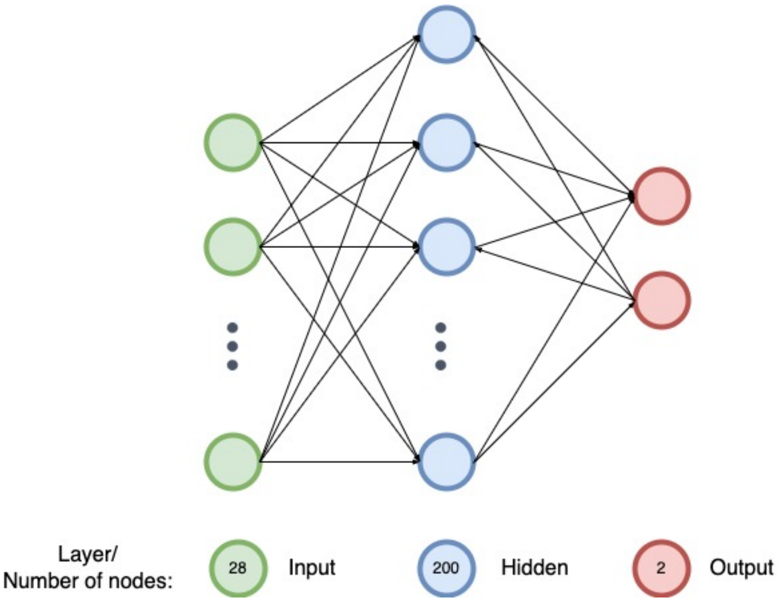


Figure 20: The final model for speech detection.

The final built classifier for the meal onset detector is shown in Figure 21. The architecture used was also that of an MLP. Just like the speech detector model, the input layer was a flattening layer, with 40 nodes. The model had a total of two hidden layers, with 100 nodes in each, and a dropout rate of 50%. The activation function used for the hidden layers was ReLu. The output layer had also 2 nodes, one for each class, just like the previous classifier, Softmax was used as an activation function.

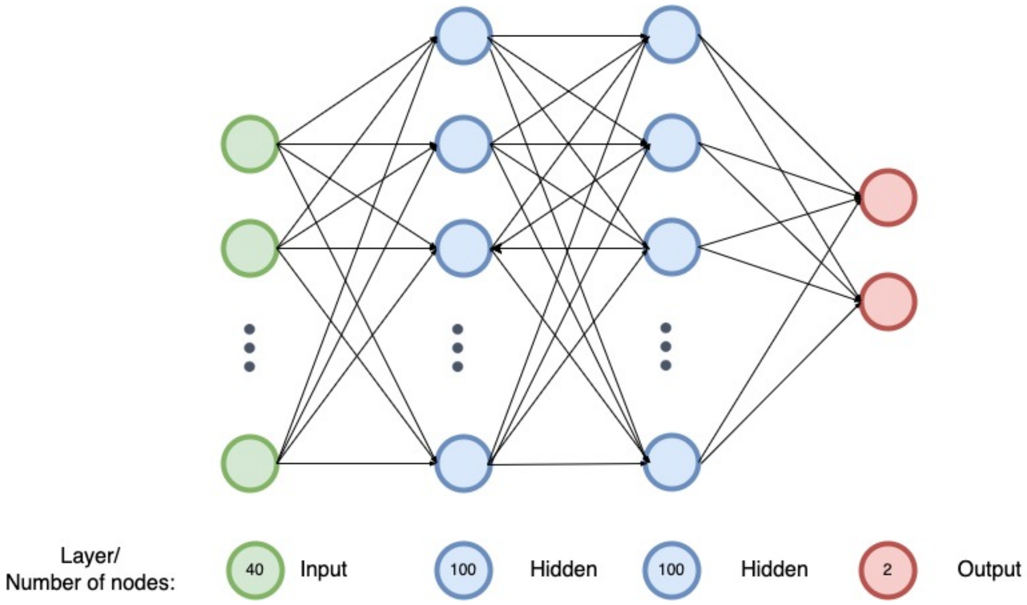


Figure 21: The final model for meal onset detection.

Each segment is assigned to a class based on the output of the models, both two output nodes give the probability of the segment being part of either one of the classes. The class with the highest probability is assigned to that segment. This is done for all the segments in the test meal, which in turn creates a predicted labeling vector that could be used to evaluate the performance of the model.

4.1.6 Evaluation of the performance

In order to make sure that the best-built model was selected, the loss curve in addition to the accuracy curve for both the training and validation data was plotted. These graphs could give an indication of whether the number of epochs is good enough, or whether the system is being overtrained or undertrained. These graphs can also give an indication of whether the learning rate is too high or too low.

In addition, the final system performance was tested using the test data. The predicted labels were plotted against the true labels, this plot was used to give an idea about where in the recordings the classifier performed good or bad. In addition, the confusion matrix was used for assessing the performance of the system. Values such as precision, recall, accuracy, and F1 score were also calculated, as this gave exact values for the performance of the system on the test set.

For each time the model is trained, the initial conditions vary at random, thus the model performed differently from run to run. That is why it was necessary to run the system multiple times and make sure that the results were not just a one-time case. Thus all the results presented in the next section are stored after running the system multiple times.

4.2 Results

In this section, the results from the speech and meal onset detectors built in the previous section are presented.

4.2.1 Speech detection

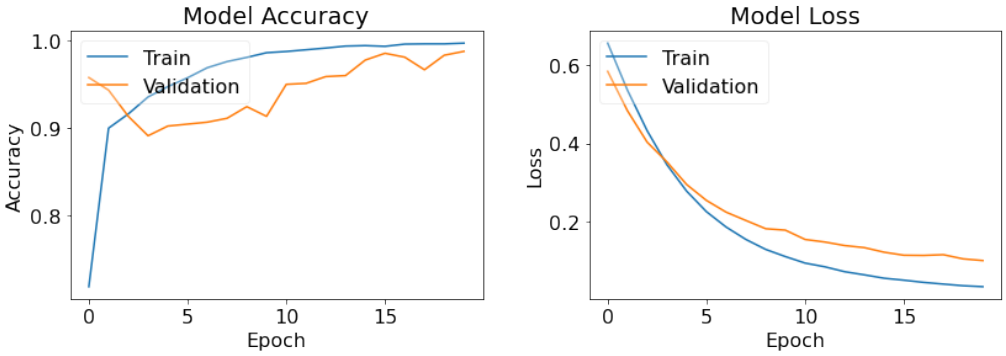
When using ANOVA F-test on the labeling vector and the features, the features that had the highest score were often the power features in the range of 0 Hz to 2 kHz. In addition, the total power feature had also a high score. As mentioned earlier, in section 4.1.5, only 28 features were selected, thus only those with the highest score were used to train and test the classifier.

For the final built system, the features extracted using a bandpass filter, with a segment length of 10 s gave the best results. This is clear when looking at the results shown in Table 2. In this table, all the extracted results from the confusion matrix are represented for all the tested feature extraction combinations. The best feature extraction combination was used to extract all the results represented below in this section.

When it comes to the final built classifier, it had a precision, accuracy, recall, and an F1 score of 99% on the test data. The classification system had an accuracy and loss curve per epoch for the training and validation set as seen in Figure 22. As could be seen, the model was run for a total of 20 epochs, as usually after that the model showed a tendency to be overtrained. It could be seen that the accuracy for both the training and validation set was in the range of 95% to 97%, while the loss was is in the range of 0.1 to 0.15. The final built model had a learning rate of 10^{-3}, and used the whole training data as the batch size.

Filtering	Segment	Precision	Recall	Accuracy	F1-score	TN	FN	FP	TP
Bandpass	10 s	99 %	99 %	99 %	99 %	1100	10	4	350
Lowpass	10 s	97 %	96 %	96 %	96 %	1100	18	35	340
Bandpass	20 s	96 %	95 %	95 %	95 %	540	7	29	170
Bandpass	5 s	97 %	97 %	97 %	97 %	2200	14	87	710

Table 2: Table showing the result from training and testing the model with different features.

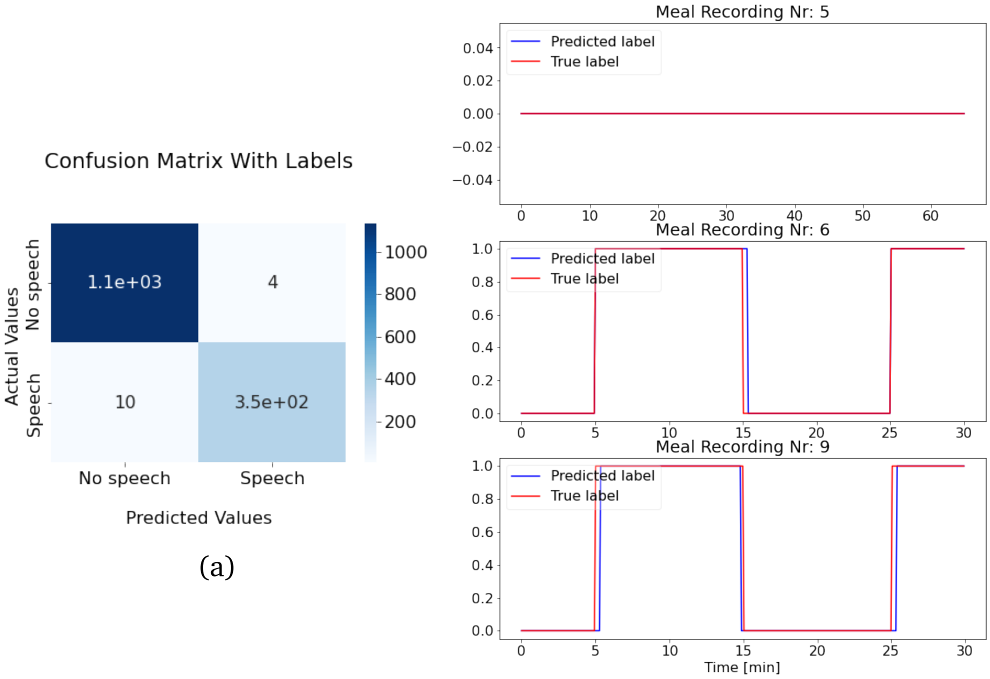


(a) (b)

Figure 22: a) Accuracy and b) Loss curve per epoch for the training and validation set using the final model.

The confusion matrix for the test data could be seen in Figure 23 a). From the confusion matrix, it was clear that the system performed extremely well, and it did not struggle to distinguish between speech and non-speech segments in the meal recordings, except for a couple of times, 14 to be precise.

When looking at the predicted labeling vector, as shown in Figure 23 b). It could be seen that for the first meal, there was no speech, and this was classified correctly by the model, that recording had only meal onset. As for meal recording numbers 6, and 9, both recordings had speech between the 5'th min to 10'th min and the 25'th min to 30'th min. These two recordings had also some speech during the meal onset, between the 15'th min to 20'th min, but as could be seen in the figure this wasn't picked up by the detector at all. Most FPs and FNs seem to be either at the beginning or at the end of the speech regions.



(a)

(b)

Figure 23: a) Confusion matrix and b) true and predicted labeling vector plotted over time for the test set using the final model. Label "1" is speech, while label "0" is no speech. For most of the meal recording, the true and predicted labels (labeling vectors) are overlapping.

4.2.2 Meal onset detection

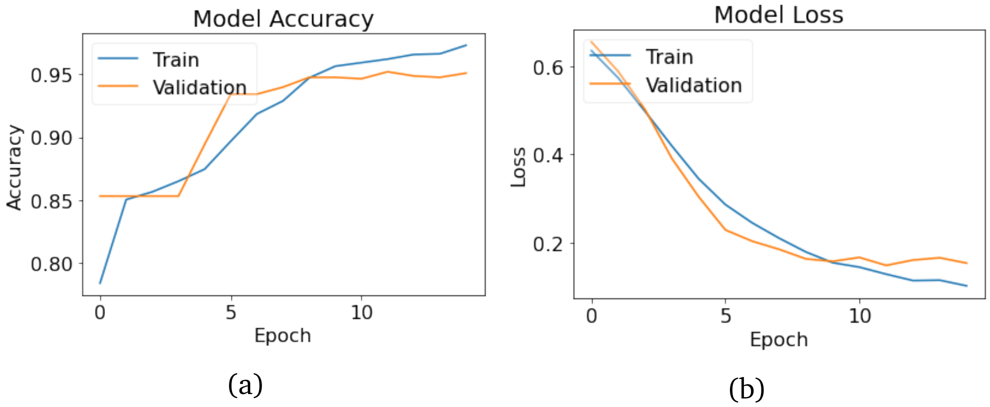
The most selected feature for this classifier was the power feature in the 0 Hz to 100 Hz band, this feature had much higher score than all the other features. The other most selected features were the power and power fraction features in the frequency range of 0 Hz to 1 kHz. For this classifier, only the 40 best features were used, as mentioned earlier, section 4.1.5.

For this system, the recording data that performed the best were the lowpass filtered with 10 s segment length, as could be seen in Table 3. In the table, all the metrics extracted from the confusion matrix are represented. The best feature extraction combination was used to extract all the results shown below in this section.

Filtering	Segment	Precision	Recall	Accuracy	F1-score	TN	FN	FP	TP
Bandpass	10 s	87 %	88 %	88 %	88 %	1200	110	71	73
Lowpass	10 s	96 %	95 %	95 %	95 %	1300	10	64	170
Lowpass	20 s	92 %	91 %	91 %	88 %	660	68	0	22
Lowpass	5 s	92 %	91 %	91 %	88 %	2600	170	36	190

Table 3: Table showing the result from training and testing the model with different features.

The built classifier using these features had a precision of 96%, and recall of 95%, while the accuracy and F1 score were both 95% on the test set. The final meal onset detection classifier performed similarly to the speech detection classifier, as could be seen in the learning curves for both the training and the validation set, Figure 24. The train and validation set accuracy was around 95% to 96%, while the loss was in the range of 0.1 to 0.2. The final model had a learning rate of 10^{-3} and used the whole training data as the batch size.

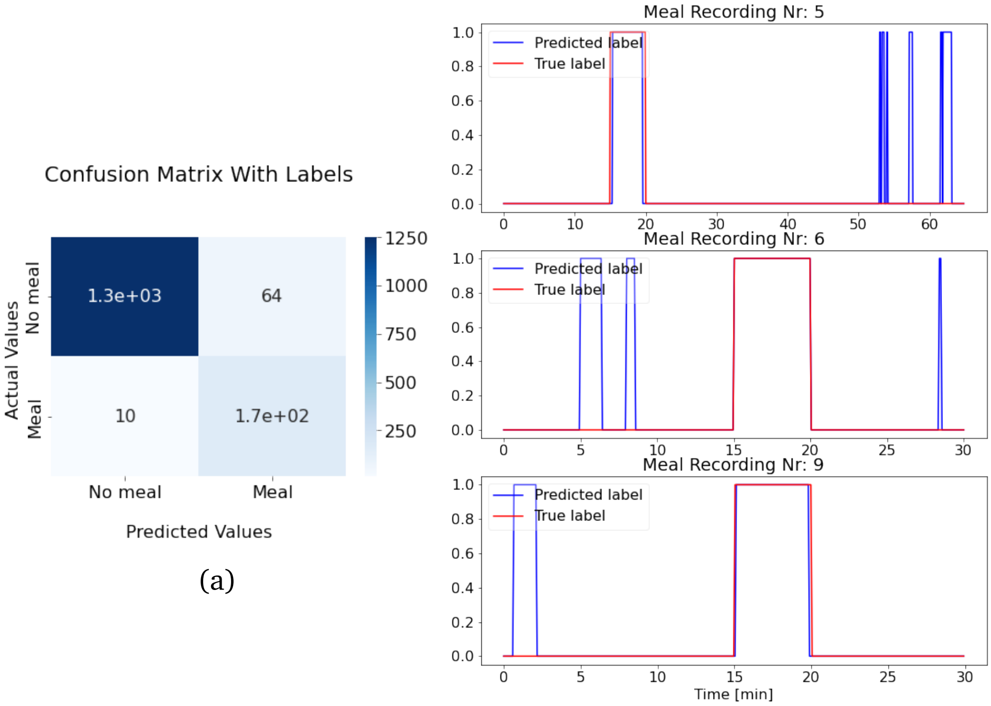


(a)

(b)

Figure 24: a) Accuracy and b) Loss curve per epoch for the training and validation set using the final model.

The results on the test set using the best feature extraction combination were also good, as could be seen in the confusion matrix, Figure 25 a). The results are quite similar to the speech detectors, however, there seems to be a larger number of FP's. Looking at Figure 25 b), for the 5'th recording, it's clear that most of the FP's appear at the end of the recording, as for the 6'th recording most of the FP's appear to be at the region of the recording where there is speech. The 9'th recording had a large number of FP's showing at the beginning of the recording. All of these contributed to the high number of FP's (64). The average meal onset detection time for this classifier is 5 min, as long as the removable noise segments are accounted for, such as speech.



(a)

(b)

Figure 25: a) Confusion matrix and b) true and predicted labeling vector plotted over time for the test set. Label "1" is meal onset, while label "0" is no meal onset. For most of the meal recording, the true and predicted labels (labeling vectors) are overlapping.

4.3 Discussion

In this part of the thesis, the implementation methods and results represented in the previous sections will be discussed.

4.3.1 Speech detection

The bandpass filtered features gave the best results for this detector, this might be due to the fact that noise in the lower frequency range of 0 Hz to 100 Hz had high power. Since PSD features were used, the noise level affected the feature selection algorithm, ANOVA F-test, such that when lowpass filtered features were used, the most selected feature was often the feature in the 0 Hz to 100 Hz band, this was not the case for the bandpass filtered features.

The most selected features from the ANOVA F-test algorithm were always the total power, and the lower frequency PSD features, 0 Hz to 2 kHz. These features had the most power when compared to other frequencies. There was also a clear difference in power between speech regions of the recording and non-speech regions of the recording as shown in Figure 18. All of this helped with the good performance of the classifier.

The results represented in Table 2 show that increasing the segment length from 10 s to 20 s leads to more FP's, especially in the beginning and the end parts of the speech regions in the recording. This could be because, increasing the segment length leads to picking up more of the non-speech parts, especially in the first 2-3 segments that follow before and after the speech regions. Hence more FP's were picked. Reducing the segment length from 10 s to 5 s led also to more FP, however, unlike the 20 s segment, most of the FP were within the speech region. This could be because, in the speech regions, the subject is not reading non-stop, thus some parts might have no talking. This is picked up more often in the shorter segment length, thus more FP's are given.

It's worth noting that the few FP's for the bandpass 10 s segment length features were always at the beginning or at the end of the speech region, as shown in Figure 23. This could be due to the fact that the labels in the labeling vector were set to the nearest minute, however, the subject did not always start reading on the dot. Hence the labeling of the speech region is not perfect. This is clearly picked up by the detector.

The final built model shown in Figure 20 used only a single hidden layer. This was because, during the early stages of the model it seemed that adding more hidden layers just added unnecessary complexity, and the result did not improve or change by much. That is why the model was kept as simple as possible. Changing the number of nodes in the hidden layer did not have a substantial effect on the result, that is why it was kept at 200. As for dropout, the value was selected based on the loss curve, in order to avoid overtraining the model.

The model's output layer used Softmax instead of Sigmoid as an activation function, even though the classification problem was binary. This was due to the fact that the model performed better with Softmax, probably due to having one extra output node, and the fact that softmax outputs are interrelated. Even though the additional node at the output increased the complexity of the model, the additional gradients constrained the network and prevented overfitting.

When it comes to the learning rate, it seemed that the network learned pretty fast, even though the whole training data was used as the batch size. It could be seen in the accuracy curve plot, Figure 22 a), that there are a lot of fluctuations in the accuracy in the first few epochs, however, it stabilizes more near the 10'th to 20'th epoch. This might have been due to having a large learning rate, 10^{-3}. However, when looking at the loss curve, Figure 22 b), it looked more stable. Increasing the learning rate led sometimes to a more stable accuracy increase on the validation set, but usually, the validation loss became worse, and more epochs were then needed to train the model. That is why the learning rate was kept as it is.

The results from the confusion matrix and labeling vector plots, Figure 23, showed that speech detection was possible. The detector was not confused by the meal onset region at all. A big contributing factor to that was that speech usually had a much higher power for lower, and higher frequencies than for the meal onset. The difference in power between the meal onset and speech regions of the recordings could be clearly seen in Figure 18.

Two of the test recordings, meal number 6 and 9, had some speech during the meal region onset, between the 15'th min to 20'th min. This was not detected at all by the system, Figure 23 b). This is not a problem for now, as the system was only intended to detect speech outside the meal region, as it was important to be able to separate those two regions. This could be detected by the other built systems later on, such as the swallowing detector. This should be investigated later on, as those speech regions inside the meal onset could induce FP's, which in turn affects the performance of the system.

Unlike the SVM result from the previous project [7], the built classifier was able to clearly separate speech regions in the recordings from non-speech regions. This showed that ANN models are more powerful and capable than SVM, for such a problem, given the available data set. The plan of using this system was to remove most speech-related noise that could affect the food intake monitoring, this seemed feasible using this classification model.

4.3.2 Meal onset detection

Unlike the speech detector, the lowpass filtered features performed the best for the meal onset detector. The most selected feature using the ANOVA F-test algorithm for this model was the 0 Hz to 100 Hz power band feature. This frequency region had always high power, especially during the meal onset. Bandpass features did not give good results, since most frequencies in that region were filtered out, and thus some important meal information was lost.

The results in Table 3 showed that increasing the segment length from 10 s to 20 s lead to having no FP's outside the meal region, however inside the meal region there was a lot of FN, and the classifier did not work properly for the 20 s segmentation length. When decreasing the segment length from 10 s to 5 s, the number of FP's increased, as more noise was classified as meal onset. Also, there were a lot more FN's inside the meal region. The classifier shifted a lot between the class "1" and "0", meal and no meal, during the meal onset. This is similar to the speech detector case, as the subjects are not consuming food non-stop in the meal region. Thus some segments might be missing the typical characteristics of a meal segment, which explains the high number of FNs.

The final built model for the meal onset detector, Figure 21, needed more hidden layers than the speech detector, Figure 20, since the problem was more challenging with regard to the overlap in frequency content. The frequency range for speech and other noise sources in the recording such as heartbeat, breathing, and talking overlapped with the lower frequency range of the meal onset, due to lowpass filtering. Having 100 nodes in each layer seemed to work pretty well, changing it did not really affect the performance or the training time by much. In addition, the learning rate of the model, which was 10^{-3} worked fine, as the model was trained fairly quickly, even though the whole training set was as the batch size. In the output layer, Softmax was also used instead of Sigmoid as an activation function. The reasoning behind this choice is similar to that of the speech detection classifier, mentioned in the section above.

The results from the confusion matrix and labeling vector plot, Figure 25, showed that the model suffered from a few FP's. Most of the FP's were concentrated at the beginning and end of the recordings. The FP's at the end of the recordings are mostly due to friction noise in the recordings, as subjects moved a lot more during the end of the recordings, due to stiffness in the muscles from prolonged sitting. The noise at the beginning of meal recording nr. 9 seems to be caused by the heartbeat noise in the recording. The heartbeat rhythm was faster at the beginning of the recording but went down later on. This showed that this type of noise is harder to remove than other noises such as speech, as there is plenty of overlap in the frequency domain. The noise in the 6'th can be easily removed by applying the speech detector from earlier, as this noise is caused by talking. This noise was easily detected by the speech detector and is not a big problem.

The average meal onset detection time was 5 min when accounting for the removable noise, such as speech that could be removed using the speech detector. This was quite the improvement when compared to the CGM system's time delays of 30 min to 40 min. Given the high accuracy, and F1 score, the duration of most meals was classified correctly. The system could be feasible for automatic meal onset detection only if the induced FPs due to noise can be removed. This could be possible by using a specific meal duration threshold, where for example consecutive labels of duration shorter than 1 min are removed, or something similar. This must be investigated if such a system is to be used for automatic meal onset detection to help CGM systems with the time delays, and when insulin should be administrated.

This system was originally only intended to detect the meal onset region in order to decide when insulin should be administrated. Furthermore, it was used to help with building meal type and swallowing detectors that are introduced in the next section. When training the other classification systems, if the whole recording is used, then the training time and complexity of the model is increased. That is why instead of training the meal type detector and swallowing detector on the whole recording, only the meal onset region is used. The meal onset detector could be used to pinpoint the start and end of the meal, in addition, the speech detector system could be used to remove the noise from talking if needed.

The meal onset detector built using ANN performed much better than the previous project's SVM classifier. The SVM classifier from the previous project had a recall of 85% [7], while the ANN classifier had a recall of 95% on the test set. Even though the systems were built using different methods, more or less the same type of data and features were used, and it was clear that ANN performed much better.

5 Swallowing and meal type detector

This section introduces how the swallowing and meal type detectors were built using a CNN classification model. The main idea behind these two detection systems was to see whether it was possible to build a system that gave an estimate of how much food is consumed during the meal onset. A swallowing detector system could give an idea about the amount of food by counting the number of swallows during the meal onset. By combining the results from a swallowing detector and a meal type detector, the amount of food consumed could be roughly estimated, as the more liquidy the food consistency is the more swallows occurs within a specific time period, and the drier the food is the fewer swallows occurs. This is due to fact that more chewing is required the dryer the food is so that the food becomes moist enough before swallowing.

5.1 Method description and implementation

This section of the project provides a step-by-step description of how the models were implemented. In addition, the reasoning behind some of the steps is briefly discussed.

5.1.1 Data acquisition

For this part of the project, a total of 20 new recordings were acquired, using the protocol and recording procedure described in sections 3.3 and 3.5. From each meal type, 10 recordings were acquired. These recordings were acquired using the same two subjects from the previous project [7]. Both of the subjects were available for 10 recording sessions, 5 oat and 5 salad meal recordings were acquired from each subject. Only these recordings were used for this part of the project, as the 10 recordings from the previous project [7] did not have the same protocol, also the meals consumed were different.

5.1.2 Labeling the data

After acquiring the data, the recordings were then labeled manually one at a time using Audacity. All the swallows during the meal region were labeled, and both the beginning and the end of the swallows were marked manually by listening to the recordings. A typical swallow, taken from the row recordings (salad recording) is shown in Figure 26, where the dotted blue lines show the beginning and the end of a swallow. The three phases of a swallow are clearly visible in the plot, the first phase, IDS, is from 0.22 s to 0.45 s, the second phase, BTS, is from 0.45 s to 0.7 s, while the final phase, FDS, is from 0.7 s to 0.85 s. These three phases could be clearly identified for most swallows, even though the swallow duration varied.

Originally the swallowing sound recordings were used to label the swallows in the recording, however, this was much harder to do than expected, as breathing and heartbeat sound affected the noise levels in the recordings. Hence at times, it was difficult to hear the swallow. Instead, the recordings intended to capture chewing sounds were used to label the swallows, since the recordings were acquired simultaneously, the recordings were synchronized and the swallow locations corresponded in both recordings. Using these recordings, it was much easier to label the swallows as the noise from breathing and heartbeat was not as apparent in these recordings.

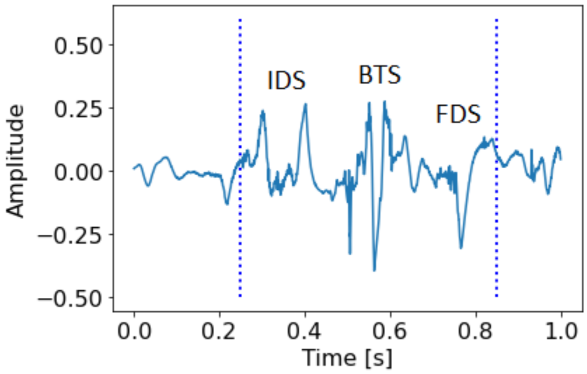


Figure 26: Raw swallow sound sample extracted from a salad meal recording, the swallow begins and ends between the dotted blue lines. The three phases of a swallow are also clearly visible.

5.1.3 Data pre-processing

Just like the speech and meal onset detectors, the pre-processing steps described here are similar. First, the recording data were quantized, to reduce the bit rate from 24 bits to 16 bits. After quantizing, the data were decimated, this was a two-step process. First, the data were lowpass filtered, since this part of the project is about extracting meal information, all the frequency ranges that could contain information about the meal were kept. Hence Bandpass filtering was not used in this part of the project. The second step of decimation was downsampling the data, the data were downsampled to 8 kHz, by using a downsampling factor of $L = 6$.

5.1.4 Feature extraction

The idea early on was to once again use PSD features to build the models, however, this did not work as planned. Using PSD features, both the built models did not work, since power features were too simple and were not able to capture the relevant meal type, and swallowing information. Thus other features had to be used. With regard to the available research regarding swallowing detectors, MFCC and Mel spectrogram seemed to be commonly used, hence both features were tested. MFCC features and its Delta features were tested early on, and were calculated in a similar way to PSD features and used to build a feature matrix like the one shown in Figure 19. However, once again the results were bad and thus these features were also dropped. Mel spectrogram showed some promising results, thus only Mel spectrogram features were extracted for this part of the project.

Before extracting Mel spectrogram features, each meal recording was first normalized individually, by simply dividing with the max value within the data. Since the plan was to use only the meal onset region for training and testing the model, only the meal onset region was kept after normalization. After normalizing the data, the data was then segmented using three different segment lengths, 0.6 s, 1.0 s, and 1.5 s, with a hop length of 0.2 s for all the three-segment lengths. Hop length is the region of the segments that do not overlap. The segment lengths were selected based on the typical length of a swallow, while the hop length was selected based on what gave good results.

During labeling the average swallow length of oats and salad was about 0.6 s, some swallows, however, could reach a duration of up to 1 s, thus these two segments were used. In addition, 1.5 s was used to capture any relevant information in the region before and after a swallow. A box plot of the labeled duration of oat and salad swallows is shown in Figure 27, as could be seen oat bolus seemed to have a longer swallowing duration on average than salad bolus.

After segmentation, each segment was framed into frames of lengths 0.2 s, 0.1 s, and 0.04 s, with a respective frame hop length of 0.1 s, 0.05 s, and 0.01 s. These values were selected to determine whether a higher or lower time resolution works the best, and also to test a wide variety of values. For all the frames in a segment, Mel spectrogram features were calculated, in total 128 features were calculated for each frame. Having this different segmentation, framing, and hop lengths allowed for more testing with regard to how the performance

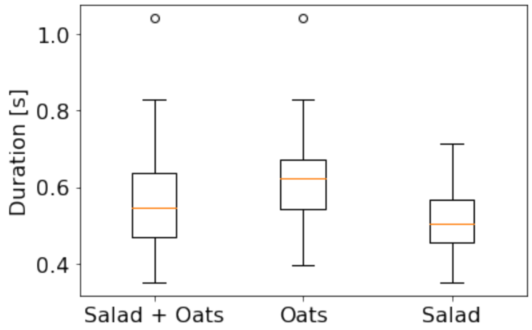


Figure 27: Box plot of the duration of the labeled swallows for all the 20 meal type recordings.

of the built classifiers was affected. The time resolution of the features is affected by the frame length, since the smaller the frame length is the better the Mel spectrogram time resolution becomes, this might have an effect on the classifiers. After this step, each feature within each segment was then normalized linearly, using (1). This scaled the features to values between 0 and 1, before saving the Mel spectrogram features as images for training and testing the model.

Originally for both the swallowing detector and meal type detector, the features were extracted from the whole meal onset region, however, this was later on changed for the swallowing detector. Feeding the swallowing detector whole meals during training did not lead to good results, as creating a labeling vector for the data was much harder like this, since some segments included both parts of a swallow and a non-swallow. For this reason, the features for training and validation set for the swallowing detector were acquired in a different manner.

The manually labeled swallow markings using Audacity were used to extract only the swallows. Swallowing segments were extracted by using the beginning of a labeled swallow marking and taking the following samples until there is enough samples to fill the desired segment length. For each swallow segment, Mel spectrogram features were extracted. Similar framing length and frame hop length was used as mentioned earlier. The non-swallowing segments were then extracted just like earlier, however now a constraint of 0.5 s before and after each labeled swallow was applied. This constraint was added to avoid extracting segments that included parts of a swallow. This reduced the total amount of training segments in each recording, however, the labeling vector became more accurate.

These extracted swallow segments contained most of the time a whole swallow, in addition to a few non-swallow parts before and after the swallow, this depended on the segment length. The 0.6 s swallow segments included most of the time a whole swallow. The 1.0 s segments included 0.2 s before a swallow, a whole swallow, and usually, in the end, it included some non-swallow parts. The 1.5 s segments, included 0.5 s worth of samples before a swallow, then the whole swallow, and also non-swallow parts at the end. The swallow segment contents are illustrated in Figure 28. An example of a 1.0 s segment swallow Mel spectrogram is shown in Figure 29, for both oat and salad swallows.

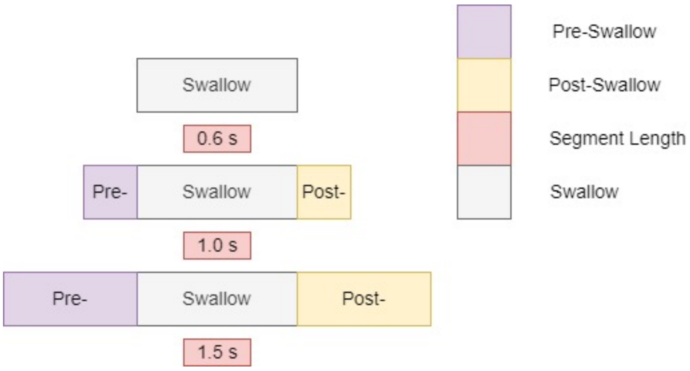
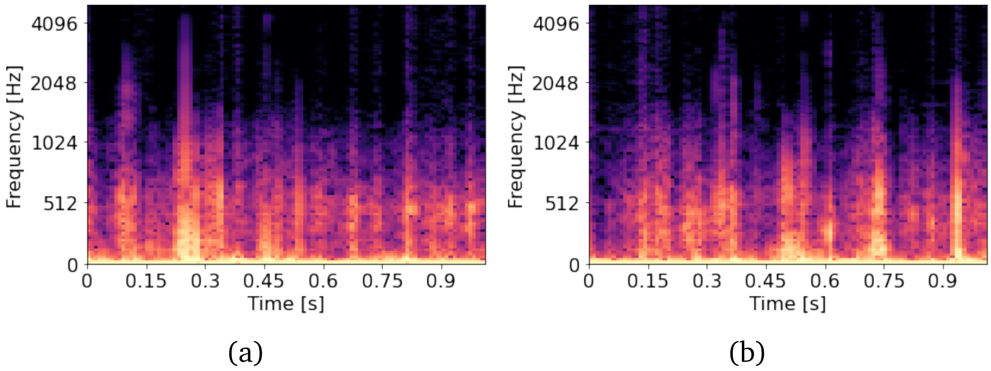


Figure 28: Illustration of the extracted swallow segments, the Figure shows the content of the three used segment lengths. Pre- and Post-Swallow are the parts in the recording that follow before and after a swallow.



(a) (b)

Figure 29: Mel spectrogram of swallowing a) oat bolus and b) salad bolus. The segment length used is 1.0 s, with a frame length of 0.04 s, and a hop frame length of 0.01 s. Both swallows start and end around 0.2 s and 0.8 s.

5.1.5 Data splitting

For both systems, all of the 20 meal recordings data were used for training and testing the models. 14 of the meals were used for training, 2 were used for validation and 4 were used for testing the model. The meals assigned for each data type were picked at random.

Since the swallowing detector used more accurately labeled swallow segments for training and validation, there were a lot fewer segments to train on. On average there were about 24 swallows in the meal onset duration for the salad meals, and 19 swallows for the oatmeals. There were on average 400-450 non-swallow segments acquired from each recording. Since there were only 19 to 24 swallow segments and about 400 to 450 non-swallow segments in each recording, this unbalance in the data affected the performance of the detector. For this reason, only 25 non-swallow segments were used from each recording in the training and validation set, these were selected at random. Meanwhile, all the swallow segments were used from each recording in the training and validation set. This allowed the training and validation data to be more balanced with regard to the data available for each class.

5.1.6 Training and classification

The first step before training the classifier is once again to create the labeling vector. For the meal type detector, two different labeling vectors were tested, and thus two detectors were also built. The first labeling vector had three classes, the segments in the meal onset region that had no swallows were either assigned a label of "1" or "2" depending on the meal consumed, oats being class "1" and salad being class "2". The swallow segments were assigned the label "0".

As for the second labeling vector-only two classes were used, all the segments in the oatmeal data were assigned the label "0", while all the segments in the salad meal data were assigned the label "1". Two labeling vectors were created to see whether more accurate labeling of all segments was helpful or not. Only one labeling vector was built for the swallowing detector. The labeling vector had two classes, the swallowing segments were given a label of "1" and non-swallowing segments were given a label of "0".

After building the labeling vector, different ANN models (MLP and CNN) were built and tested on the data that was segmented using 1.5 s, and 0.2 s hop length, with a framing length of 0.2 s and frame hop length of 0.1 s. This data was used as a reference for building the models. The model was built in a similar way to that of the meal onset and speech detector, by trying and implementing different parts of the model, one step at a time. It started by testing a different number of nodes, then changing the number of hidden layers and their activation functions, and finally choosing a proper learning rate, number of epochs, and batch size.

When a model that had good results was built, all the other segment lengths were then tested before settling on a specific segment length based on the best results. When the best segment length is selected, the other proposed framing and hop length combinations are then tested to see which one gives the best results. When one segment, frame, and frame hop length combination is selected, the model is tuned one final time to assure that the best model for the data is selected. This step was the same for both swallowing and meal type detector.

For the swallowing detector model, the final built classifier is shown in Figure 30. The final architecture was that of a CNN model since it usually works the best for image classification problems. The CNN consisted of only one convolution layer, the input layer, that used ReLu as an activation function. 64 filters were used in the convolutional layer, and the kernel size was 2x2. The input data was in the shape nx5x128x1, where n is the number of images, and 5x128x1 is the dimension of the 2D images converted into 3D since convolution layers require 3D inputs.

The shape of the data is converted to nx4x127x64 after the convolution layer, due to the filters (64 features), and the convolution operation. After the convolutions layer followed a max-pooling layer, with a kernel size of 2x2. The data was then flattened for use in fully connected layers. After the flattening layer, there were two hidden layers, each having 64 nodes, with a dropout rate of 50%. Both layers used ReLu as an activation function. Lastly, the output layer had 2 nodes, and used Softmax as an activation function, since the results were better this way, even though the classification problem is binary.

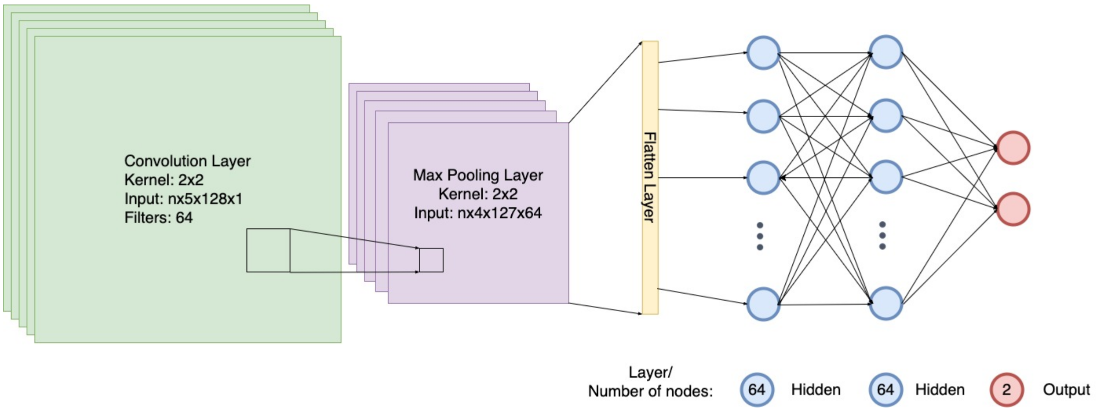


Figure 30: The final model for swallowing detection.

The final classifier for the meal type detector is shown in Figure 31. The final architecture was also that of a CNN model. The model consisted of one convolutional layer, the input layer, that used also ReLu as an activation function. The convolutional layer had 64 filters, and a kernel size of 3x3. The input size was nx9x128x1, where n is the number of images used for classification, and 9x128x1 is the dimension of the images. After the convolution layer, a max-pooling layer was used with a kernel size of 3x3. The output of that layer was flattened before being sent to fully connected layers. There were 3 hidden layers in total, all of them used ReLu as an activation function, and had 128 nodes. The first and third layers used a dropout rate of 30% while the second layer used a dropout rate of 50%.

Since there were 2 labeling vectors, one with three classes, and one with two classes, two classification models were also used. Both models had the same architecture, the same input layer, and hidden layers structure, the only difference was the output layer. When the labeling vector with three classes is used, the output layer used has 3 nodes, one for each class, just as shown in Figure 31. Since this is a multi-class classification problem, Softmax is used as an activation function for that model. When the labeling vector with 2 classes is used, the number of nodes in the output layer is reduced to 2, and the activation function is kept as Softmax. This was due to the fact that the results were better this way, even though the problem is that of binary classification.

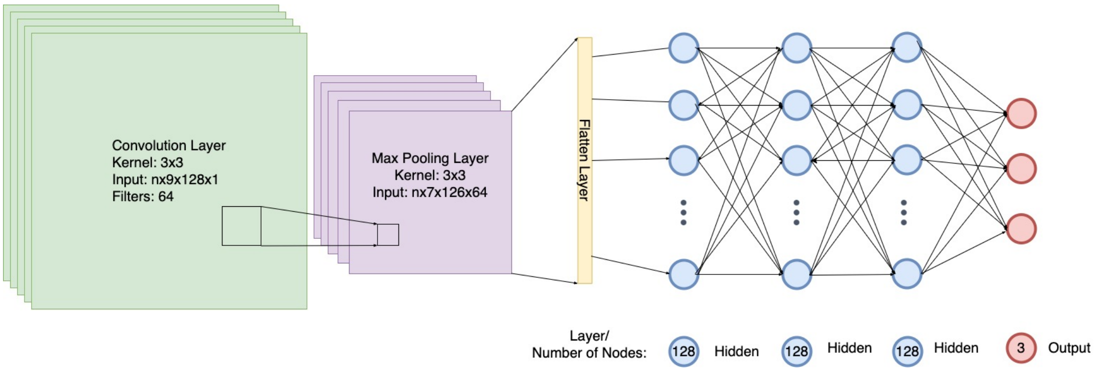


Figure 31: The final model for meal type detection.

5.1.7 Evaluation of the performance

The performance of both the meal type and swallowing detector was evaluated in a similar way to the meal onset and speech detector systems. The model's performance on the training and validation set was first evaluated using the learning curves, the accuracy, and loss curves. The performance on the test set was then evaluated using the confusion matrix, where values such as precision, recall, accuracy, and F1 score were calculated. Also, a plot showing the predicted and true labeling vectors was used. In addition, since the initial conditions vary at random for each iteration, the systems were run multiple times before storing the results.

Using only the confusion matrix values could be misleading at times for evaluating the performances of the meal type and swallowing detector, hence additional measurements were included. For the swallowing detector the number of correctly classified swallows, in addition to the number of falsely predicted swallows were counted. These were counted by summing the amount of available consecutive segments labeled as a swallow, segments assigned the label "1" in the predicted labeling vector. If those consecutive segments lay within an actual swallow in the labeling vector plot then it is counted as a correctly classified swallow, otherwise, the consecutive segments are counted as a falsely predicted swallow.

When it comes to the meal type detector, the certainty of the classification was calculated, which is basically the ratio of correctly classified segments within a meal recording data. Certainty was calculated by simply counting the number of correctly classified segments and dividing it by the actual total number of segments for each test meal recording, (15).

$$\text{Certainty} = \frac{\text{Total correctly predicted segments for a given meal}}{\text{Total number of segments for a given meal}} \tag{15}$$

5.2 Results

5.2.1 Swallowing detector

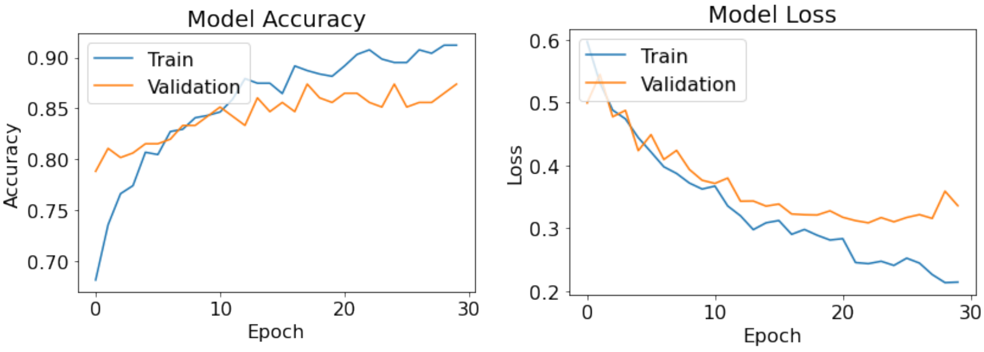
The features that performed the best, were those that were extracted using a segment length of 0.6 s, hop length of 0.2 s, frame length of 0.2 s, and a frame hop length of 0.1 s. All the results for the different segments, frames, and hop lengths extracted from the confusion matrix using the final model are summarized in Table 4. The best feature extraction method was used to extract all the results represented below in this section.

Segment	Frame	Frame hop length	Precision	Recall	Accuracy	F1 score	TN	FN	FP	TP
1.5 s	0.2 s	0.1 s	88 %	89 %	89 %	89 %	5700	390	280	11
1 s	0.2 s	0.1 s	91 %	92 %	92 %	91 %	5700	300	220	99
0.6 s	0.2 s	0.1 s	92 %	93 %	93 %	92 %	5500	300	110	98
0.6 s	0.1 s	0.05 s	90 %	90 %	90 %	90 %	5600	320	340	84
0.6 s	0.04 s	0.01 s	90 %	89 %	89 %	90 %	5600	300	410	110

Table 4: Table showing the result from training and testing the model with different features. Hop length for all the segments is 0.2 s.

As could be seen in Table 4, for the final model, the precision and F1 score was

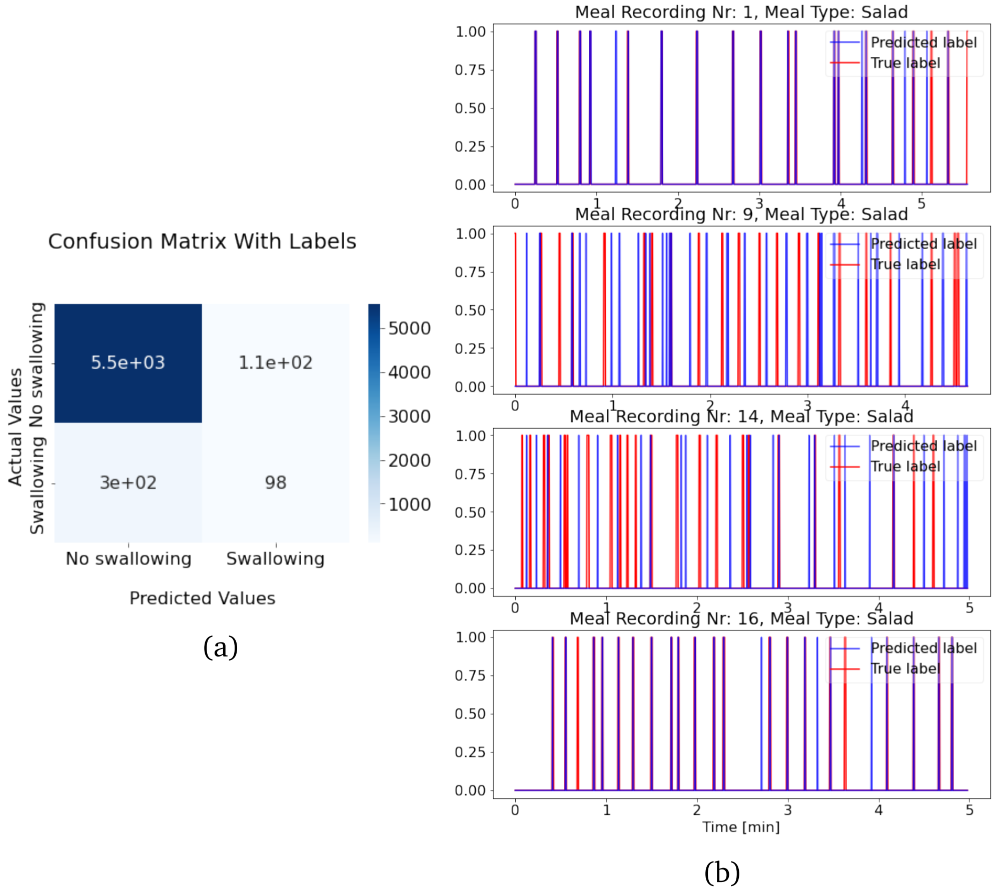
92%, while the recall and, accuracy was 93% for the test set. The final model had a learning rate of $2 \cdot 10^{-2}$ and was trained for a total of 30 epochs, with a batch size of 50, this resulted in the following learning curves, Figure 32. As could be seen the accuracy was in the range of 85% to 90% for both the training and validation set. The loss for the training set was as low as 0.22 and as high as 0.35 for the validation set.



(a) (b)

Figure 32: a) Accuracy and b) Loss curve per epoch for the training and validation set using the final model.

The performance of the model on the test set could be understood by looking at the confusion matrix and labeling plot in Figure 33. As could be seen only 1/4 of the swallowing segments were detected correctly, also a similar amount of FP was counted as TP, 110, and 98 respectively. When looking at the labeling plot, the red lines represent the true swallowing segments, while the blue lines represent the predicted swallowing segments. Purple lines indicate that the swallowing segments and predicted swallowing segments overlap. The 9'th and 14'th recordings in the labeling plot seemed to have a lot more FPs than the other two meal recordings, the 1'st, and 16'th.



(a)

(b)

Figure 33: a) Confusion matrix and b) true and predicted labeling vector plotted over time for the test set. Label "1" is swallowing, while label "0" is no swallowing. Most of the true and predicted labels (labeling vectors) are overlapping, and thus have the color purple.

The amount of falsely predicted and correctly predicted swallows for the test set are summarized in Table 5, for both before and after using the proposed smoothing algorithm. Before using the smoothing algorithm, for the 1'st meal, there were 19 actual swallows in the recording, and 17 out of those 19 were classified correctly. The classifier predicted 4 swallows falsely, due to FP's. The 16'th meal results were similar to the 1'st meal results, as could be seen in the Table. The two other meals performed badly, with regards to both the number of correctly and falsely classified swallows.

To remove most of the falsely predicted swallows, a smoothing algorithm was implemented. The algorithm worked in such a way that only the consecutive swallowing segments in the predicted labeling vector are kept. Thus any predicted swallow segment that is surrounded by predicted non-swallow segments is relabeled as a non-swallow segment. The idea behind this was to remove most of the FP's induced by any non-swallow-related noise, such as breathing or chewing. The results of the smoothing algorithm are shown in Table 5. As could be seen the amount of falsely detected swallows was reduced by a fair amount, while the amount of correctly classified was affected slightly but not as much as the amount of falsely predicted swallows.

Smoothing Algorithem	X	Before		After	
Meal Recording Number	Actual Number of Swallows	Corectly Classified Swallows	Falsely Predicted Swallows	Correctly Classified Swallows	Falsely Predicted Swallows
1	19	17	4	15	3
9	21	3	32	2	18
14	23	4	25	1	6
16	22	20	4	15	0

Table 5: Table showing the number of actual swallows, correctly classified swallows, and the total number of predicted swallows for the test meals.

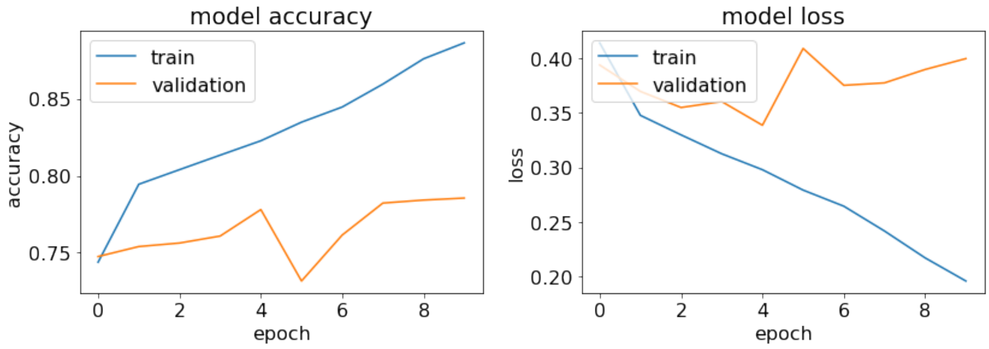
5.2.2 Meal type detector

First, the results for the meal type detector when using the multi-class labeling vector are represented. For this meal type detector, the features that performed the best were those that had a segment length of 1 s, hop length of 0.2 s, a framing length of 0.2 s, and a hop length of 0.1 s. The precision was 77%, recall and accuracy was 81% while the F1 score was 78%. This could be clearly seen in Table 6, where all the results for the different segments, frames, and frame hop lengths extracted from the confusion matrix using the final model are represented. In addition, the average certainty over all the test meals is also summarized in the Table. The best feature extraction method is used to extract all the results presented below in this section.

Segment	Frame	Frame hop length	Precision	Recall	Accuracy	F1 score	Average certainty
1.5	0.2	0.1	79 %	79 %	79 %	79 %	88.25 %
1	0.2	0.1	77 %	81 %	81 %	78 %	88.75 %
0.6	0.2	0.1	72 %	64 %	64 %	63 %	68 %
1	0.1	0.05	76 %	78 %	78 %	74 %	85 %
1	0.04	0.01	72 %	69 %	69 %	70 %	80.75 %

Table 6: Table showing the result from training and testing the model with different features. Hop length for all the segments is 0.2 s.

The final model had a learning rate of 10^{-3} and was trained for a total of 10 epochs, using a batch size of 100. The learning curves could be seen in Figure 34. The training and validation set had an accuracy of 92% and 85%, and a loss of 0.2 and 0.35 respectively.



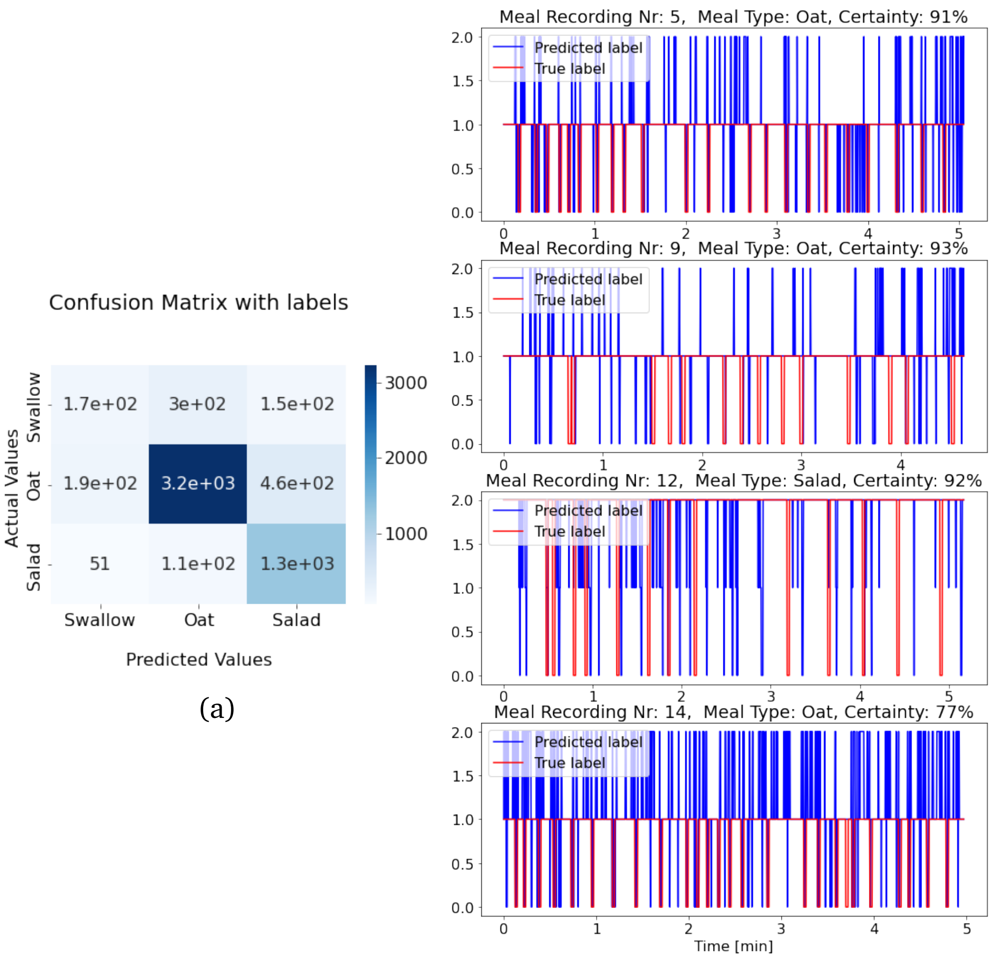
(a)

(b)

Figure 34: a) Accuracy and b) Loss curve per epoch for the training and validation set using the final model for the multi-class classification problem.

The results on the test set could be seen in the confusion matrix and labeling vector plot, Figure 35. It's clear that the results of the confusion matrix did not really provide a lot of interpretable information, the results extracted from the labeling plot were more meaningful. As could be seen the certainty for 3/4 of test meals was above 90%, and the average certainty for all the test meals was around 88.75%. Having a high certainty meant that most segments were classified correctly, and thus the labeling plot had less swinging between the classes. When looking at the confusion matrix, it seemed that oats segments are classified more often as salad, than salad segments are classified as oats.

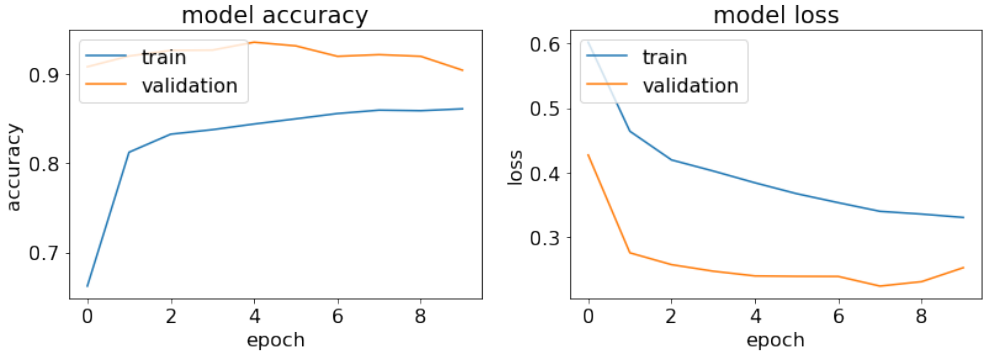
The other built meal type detector used for the binary classification labeling vector was tested using only the best performing features from the multi-class detector above. The same model was used, with the same learning rate, number of epochs, and batch size, the only difference was the number of nodes in the output layer. The learning curves, for the training and validation set, are shown in Figure 36. Unlike the learning curves of the previous model, both the accuracy and loss seemed to be more stable, as they did not fluctuate as much. The accuracy for the training set was 95%, while the validation set accuracy was around 85%. The loss for the validation set was 0.27, while the training loss was around 0.38.



(a)

(b)

Figure 35: a) Confusion matrix and b) true and predicted labeling vector plotted over time for the test set using the multi-class classification model. Label "2" is salad, while label "1" is oats, and label "0" is swallows. For most of the meal recording, the true and predicted labels (labeling vectors) are overlapping.



(a)

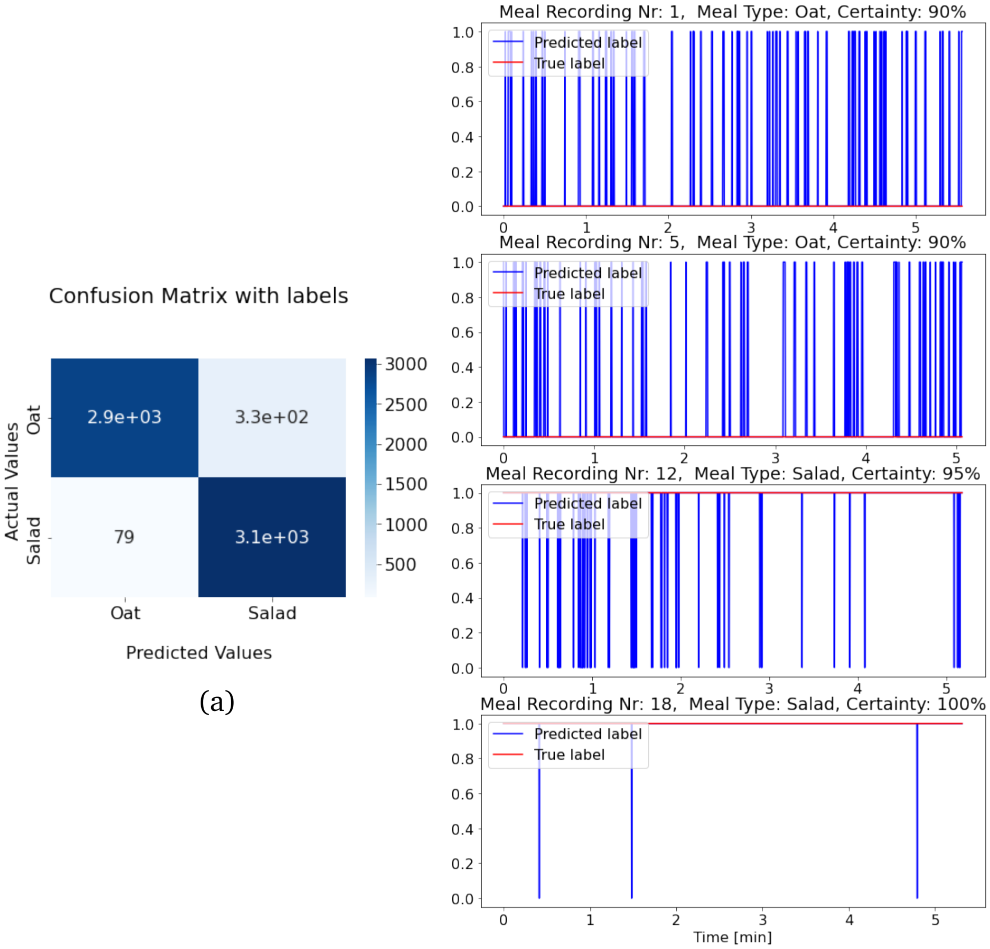
(b)

Figure 36: a) Accuracy and b) Loss curve per epoch for the training and validation set using the final model for the binary classification problem.

The model's confusion matrix and labeling vector results for the test set are shown in Figure 37. As could be seen all test meals had certainty around 90% and above, this model performed much better than the multi-class classification model. The average certainty for this model was around 93.8%, but as could be seen there were still many FP's in most of the test meals, thus the smoothing algorithm that was used for the swallowing detector was changed slightly for use in this system as well.

The algorithm was altered so that it was iterated 4 times. In the first iteration the consecutive segments, with less than two subsequent segments of the same label are relabeled to the other class. Then in the second iteration, all the consecutive segments, with less than three subsequent labels of the same type are relabeled to the opposite class. This is done two more times, for four subsequent segments, and also for five subsequent segments. Thus only the consecutive segments of the same type with more than 5 subsequent segments of the same label are left unaltered. This step removed many of the FP's.

The precision, recall, accuracy, and F1 score before using the smoothing algorithm was 94% on the test set. The precision, recall, accuracy, and F1 score after using the smoothing algorithm were increased to 96%. The average certainty increased also to 96.8%.



(a)

(b)

Figure 37: a) Confusion matrix and b) true and predicted labeling vector plotted over time for the test set using the binary classification model. Label "1" is salad, while label "0" is oats. For most of the meal recording, the true and predicted labels (labeling vectors) are overlapping.

5.3 Discussion

5.3.1 Swallowing detector

Labeling the swallows in the swallowing recordings proved to be a difficult task, as the noise level varied from one recording to another. This was due to sources of error that included the placement of the microphone which might have varied slightly from one recording to another, the movement of the subject, and whether a pressure is applied on the microphone due to slight bending of the neck or not. This made the manual labeling of the recording difficult, and for this reason, the chewing recordings were used to label the swallows. Any slight movement or bending of the neck during the meal intake did not create any noise in the recordings, and thus labeling the swallows was much easier, as it was possible to hear most of the swallows clearly.

When building the swallowing detector unlike the previous systems, not the entire meal onset region was used during training. As explained in section 5.1.5, only the swallowing segments and a random subset of the non-swallowing segments were used. This was done to improve the training quality, as previous methods of segmenting the whole region lead to some segments having parts of both a swallow and a non-swallow. This made labeling the segments more difficult, as in how much must the segment include of the swallow for it to be considered a swallow or vice versa, for this reason, the classifier did not perform well. When testing the model, however, the whole meal region was used, and the swallows could be a part of multiple consecutive segments, this had to be accounted for in the smoothing algorithm. This is why only the consecutive labeled segments of swallows, the segments labeled as "1", are accounted for as a predicted swallow.

The best features were those that were segmented with a segment length of 0.6 s, this correlated well with the average length of a swallow, which was around 0.6 s. Thus when using this segmentation length, most segments had a full swallow within. The classifier trained using the segment length of 0.6 s had the best performance when compared to other segment lengths, as shown in Table 4. When the segment length was increased, the number of FP's also increased. A reason for that is that when the segment length is increased, the training segments then do not only contain a whole swallow but also some parts of other noise sources in the recording, such as chewing and breathing. This confused the classifier and affected the training process.

Different framing and frame hop lengths were also tested for the 0.6 s segment. The results represented in Table 4 show that using a larger frame size caused the system to have a larger number of FP's, while a smaller frame size usually reduced the number of FP's. The number of TP's, however, remained more or less the same. There was no advantage in reducing the frame, or the frame hop length. The increase in the number of FP's was not beneficial, as the whole idea of the system is to estimate the amount of food consumed, based on the number of detected swallows. The more swallows are detected, the more food is assumed to be consumed, as the amount of consumed food correlates with the number of meal-related swallows.

The model accuracy and loss, Figure 32, showed a lot of fluctuation which is a sign of a high learning rate, however, the learning rate was kept as it is, as otherwise a large number of epochs was needed to train the model, and the results usually remained more or less the same. A large number of epochs usually lead to overtraining, which usually worsened the generalizability of the model on test data. The model used 50 training data as batch size, mini-batch since the results were much better and loss often converged faster. This can also affect the smoothness of the loss and accuracy curves, it usually gives smoother curves, unlike when using the whole training set as the batch size. However, this was not the case here, probably due to the learning rate.

The confusion matrix results, Figure 33 a), looked not too good at first glance, as it seemed only 1/4 of the actual swallowing segments were correctly classified, and there was also a similar number of FP's to TP's. When looking at the labeling vector plot, Figure 33 b), there seemed to be a lot of none consecutive segment's labeled as "1". These FP's correlated most of the time with the location of chews, breathing, and external sounds in the recording. This was exploited by the smoothing algorithm to remove some of the FP's, as usually such noises do not last long, and are kept within a single segment. Hence the falsely classified segments are usually surrounded by correctly classified segments. Thus only the consecutive labeled segments as "1" were kept, and every consecutive segment labeled as "1" was regarded as a swallow. Using this algorithm the number of correctly classified swallows remained more or less the same, while the number of falsely predicted swallows was reduced, as could be seen in Table 5.

This model was intended to be used for detecting the amount of meal-related swallows. When combining this information with the information received from the meal onset and meal type detector, this should then give an approximation of how much food has been consumed. Based on that information, the amount of insulin needed to be administrated could be estimated. As it stands right now for the best performing test meals (two meals), about 73% of the swallows are detected correctly, if all the test meals are included then only about 40% of the swallows are detected correctly. This is not ideal, the system needs more improvement for it to be even considered for such a purpose.

The results for the number of detected swallows for some meals were horrible, a factor contributing to these results is the quality of the recordings. Some recordings had more noise than the others. This is why labeling the swallows using the swallowing recordings was difficult. This might be due to fact that the microphone used, was originally intended to pick up bowel sound movements, thus the stethoscope-like container that was used could have affected the quality of the recordings.

Since the microphone was kept in place using double-sided tape, the pressure inside the microphone container remained constant, this allowed the microphone to pick up a lot more noise from within the body. Thus breathing, heartbeat and internal body noises had sometimes high amplitude, especially when the subject bends the neck slightly during the recordings. The fact that the microphone was also omnidirectional did not help, as uncontrolled noise from the environment was also picked sometimes by the microphone. In addition, the problems following from proximity effects, where the lower frequency content of the signal is amplified played a role in affecting the results. This is not applicable for use as it stands right now, as there will be more noise sources in a more realistic use scenario.

Sven M. Carlsen who is a professor in clinical research at the Department of Clinical and Molecular Medicine at NTNU was consulted about the possibility of using such a swallowing detector for estimating the amount of insulin. Based on his feedback, it seemed that the insulin dose is affected by many other factors in addition to the number of swallows. There are many hormones in the body that acts as insulin blockers. For example, if the patient is stressed or has been a part of some activity, more insulin will then be needed. These hormones are affected by the physical and mental state of the patient. For this reason, relying only on the swallowing sounds for determining the needed amount of insulin is not accurate enough, more measurements should be included. However, the amount of swallows is an important factor, that could help towards reaching that goal.

5.3.2 Meal type detector

Unlike the swallowing detector, which relied and was trained mainly on swallowing sounds, this detector was built using a different approach. The swallow sounds did not seem to give relevant information about the meal type, as could be seen in Figure 29, most frequency contents were similar for both the swallowed oat and salad bolus. What helped with the results were the chewing sounds captured in the swallowing recordings, the chews had higher power frequency content for salad than for oats. Since the swallow sounds did not provide relevant information about the meal type, two types of classifiers with different numbers of classes were built.

Multi-classification

The meal type detection system performed better for the higher segment lengths, as shown in Table 6. The results from the 1.5 s and 1.0 s segments were quite similar, however, the 1.0 s segment just edged it. The larger the segment is the more information is contained about the meal type, thus the classifier becomes less uncertain. It seemed that increasing the segment length led to increasing the number of TP's. Reducing the framing length and frame hop length, showed similar results to that of the swallowing detector. Reducing these values led to more FP's, and thus the systems classification performance dropped. Also, it seemed that the smaller the framing length and frame hop length is, the more the system struggled to distinguish between the oat and salad meal segments.

The model accuracy and loss, Figure 34, showed a lot of fluctuations, even though the batch size was as low as 100, which again is most probably due to the high learning rate. Lowering the learning rate could have reduced it, however, the number of epochs would have to increase to more than 10. This was not done since the training time would have increased by a lot. The training of this model already took some time due to the complexity of the model, as shown in Figure 31, and since the result was not that bad, it was kept as it is.

This model also showed a tendency to have a bias towards a certain meal type, as the certainty for the oat meal's can at times be higher than the certainty for the salad meals, and vice versa. This usually occurred when the selected meals in the training set were unbalanced, such that when the model selected more oat than salad meals for training or the other way around. This creates a bias in the training set which at times affected the performance.

The confusion matrix in Figure 35 a), did not really provide much information on how good the system actually was, this is why certainty was introduced. When looking at the average certainty for the test meals, which was around 88.75%, it was clear that the system worked pretty well.

When taking a look at the labeling plot, Figure 35 b), the systems miss-classified a number of segments throughout the meal onset. This of course is not ideal, as the whole idea of the system is to help the swallowing detector give a rough estimate of the amount of food consumed. Since the classifier is switching between labels constantly, the classifier becomes less accurate for such a purpose. For this reason, the smoothing algorithm from the swallowing detector was implemented for the multi-class classifier, however, it did not work at all. It actually worsened some of the results, due to the classifier shifting constantly between the three classes. The smoothing algorithm was only used on the binary classification model, because the smoothing algorithm worked better when two classes were used, as shown in the swallowing detector results, in section 5.2.1.

Binary classification

Using the same features and model architecture in Figure 31, with one node less in the output layer, the binary classifier was built. The learning curves seemed more stable than that of the multi-classification model, Figure 36, as the curves were smoother. There were no apparent fluctuations for the used number of epochs, given the current learning rate. The classifier performed well on the validation data, as the accuracy was higher than that of the training set. Similarly, the validation set loss was lower than that of the training set. This was not the case for the multi-class classification model, it usually had better accuracy and loss on the training set.

The result using this classifier was much better, as the certainty of the classifier was about 93.8%, and that was even before implementing the smoothing algorithm. When the smoothing algorithm was used, the algorithm worked only when it was iterated multiple times, while increasing the number of consecutive labels that are kept or changed. This worked well, as when having a low number of consecutive labels removed first, most FP's in the regions of recording with a high number of swinging between labels are removed first, the most crowded regions. When iterating with a higher number of consecutive's, the FP's in the less crowded regions of the recording are removed. This helped improve the performance, and the average certainty was increased to 96.8%.

The smoothing algorithm could be iterated for a larger number of consecutive to reach a 100% certainty rate, this is not a problem for the data in hand as of now. However, as more realistic data is included, it would be expected to have recordings of meals consisting of multiple food types. Hence, a certain limit on the number of consecutive must be selected to make detecting the food type more accurate and avoid false removal of correctly labeled segments. This must be investigated, for as it stands right now, this system might not be applicable at all unless the subject eats one food type at a time. Another possibility is to set a certain time/threshold for how often the meal type detector should detect the meal type, for example, for each 30 s, 1 min, or 2 min, this must also be investigated.

As mentioned earlier, this model was intended to be used in addition to the swallowing and meal onset detector to help identify how much food is consumed. The results from the meal onset, swallowing and meal type classifier could be used to give a rough estimate of the amount of food intake. The next step should be to try and see whether it is possible to build a mathematical model, that combines all of these results, the meal duration, percentage of swallows detected correctly, and meal type in order to estimate the amount of food consumed. Such a model was not attempted due to lack of time, in addition to the fact, that such a model will require a tremendous amount of data, which was not accessible for this project.

If such a model can be found, then it might be possible to roughly estimate the amount of food consumed, and even give an estimate about how much insulin should be administrated after the meal onset, however insulin blockers must still be accounted for. How good should both the swallowing detector and the meal type detectors be, with regard to the percentage of correctly classified swallows and the certainty of the meal type for such a model to work needs more investigation.

6 Conclusion

In this project, the feasibility of using swallowing sound recordings to improve the current treatment methods for diabetic patients was investigated. For this purpose, four different classifiers were implemented, and all of them were implemented using an ANN as a classification model. First, a speech detector was built to remove all speech-related noise in the recordings, this system had an accuracy and F1 score of 99%. Then a meal onset detector was built, to distinguish the meal region from the non-meal region in the recordings. The accuracy and F1 score using this system was 95%, while the average meal detection time was 5 min. Both these classification systems were built to continue the work done in the previous project [7], thus recordings acquired from the previous project were used to train and test both classifiers. Both classifiers were also used to reduce the complexity of the other built classifiers.

For the two other classifiers, 20 new recordings were acquired using a newly proposed protocol. Using these data, a swallowing detector and meal type detectors were built. The swallowing detector had an accuracy and F1 score of 93% and 92% respectively, while the swallowing detection rate was about 73% for the best-performing test meals. The best performing meal type detector while using a smoothing algorithm had an accuracy and F1 score of 96%, and an average certainty rate of 96.8% on the test set.

The work in this project has shown that using these proposed classification systems to give an indication about the amount of food consumed and insulin level is not enough. The needed insulin level is determined by many factors, such as the hormones that act as insulin blockers, which are affected by the physical and mental state of the patient. In addition, to estimate the amount of food consumed, the performance of the built detector must be improved, especially the swallowing detector. Thus such a system is not feasible as it stands now, however it could be used as one of the building blocks for such a system in the future since it provides relevant meal information.

7 Suggestions for future work

7.1 Test other microphones

The microphone used for the experiments might have not been ideal for collecting swallowing sounds, as it was omnidirectional. This meant that the microphone picked up sound equally from all directions, thus the microphone did not only pick up swallowing sounds but also noise from the environment. Even though the recording environment was quiet, sometimes unexpected noises are heard in the recording, such as police sirens. A more directional microphone might help with reducing the picked-up noise from the environment.

Another thing worth mentioning is that since the microphone is placed on the skin, near the sound source, the proximity effect becomes a problem. This causes amplification of the lower frequency components of the signal. This was clearly a problem as the low-frequency components of the signal had always high power. This of course could be solved by filtering the lower frequency contents using a highpass filter, or an equalizer. This must be investigated.

7.2 Improving the swallowing and meal type detector

The result of the swallowing detector was not that good, as it was affected by the quality of the recordings. The swallowing detector worked fine for the recordings that had less noise in them. If the microphone is changed then the system must be trained and tested again to see whether it's the classifier that works poorly or if it is only the recordings that are affecting it.

Also, the meal type detector should be tested for meal recordings that contain more than one meal type, since the classifier switched a lot between labels. This could affect the reliability of the model's classification. A possible approach for this problem could be dividing the meal onset region into segments of 30 s, or 1 min, or 2 min, etc, and then deciding the consistency only in that time frame. This could be done by assigning it to the label with the highest certainty. This must be looked into if such a system is considered for future work.

7.3 Collect more data with more variability

All the collected data for the models built in this project were conducted by only 2 subjects. The systems built could have been subject-dependent, but that is difficult to conclude with the amount of data at hand. Hence it is important for future work to include data collection from more subjects, from different age groups, genders, etc.

Different meals must also be included in the new data set recordings, as only having two sets of meals was fine in the first stage of the project. More common foods such as bread, rice, meat, and different types of fruits and vegetables can be included in the next stage of the project. Also, different types of liquids can also be included, such as water, soup, and soda to see whether it is possible to differentiate between such liquids or not.

In addition, recording acquired in a more realistic environment should be included. This could help with reducing the amount of FPs due to both external and internal noise sources, such as friction noise and noise from the environment. As more data is collected the uncertainties in the system's subject dependability could be reduced, in addition, the performance of the systems could also improve as more data for training and testing is provided.

7.4 Consulting a doctor

If these systems are to be used in estimating the amount of food consumed for diabetes treatment, a doctor must be consulted, with regard to how good the systems should be. For example, how large should the percentage of correctly detected swallows be for the system to be reliable for such a purpose? This, in turn, introduces a more realistic goal/criteria for the system to satisfy, with regard to the expected performance. The system should then be improved in such a way that the criteria are satisfied.

In addition, how can the information obtained from the built classifiers be used to calculate the needed insulin dose, should be looked into. 40% to 50% [62] of the daily insulin cannot be calculated using the proposed systems, because it is due to insulin being replaced overnight, background insulin replacement, and insulin between meals. The rest 60% to 50% of insulin is due to food, and high blood sugar correction, this is where the proposed systems could be used. If the insulin blocking hormones are not taken into account, how much of that percentage is covered by the built classification systems, and how reliable is it then. It's important to take that into consideration in future work.

7.5 Amount of food consumed

Another thing that must be investigated is how the amount of food consumed is gonna be estimated using the proposed systems. One possible way of doing that is collecting more data, and then using some type of classifier and solving the problem as some kind of a regression problem. ANN or other types of classifiers could be used for this problem, and it can be solved by relying on the known meal information and the results obtained from the built classifiers. How this could be approached is up for discussion.

7.6 Meal type detector

During the early stages of building a meal type detector, the system was built using the swallowing segments only. It was trained using swallowing segments that were labeled either as oats or salad, "0" or "1". This, however, did not work, due to the fact that the frequency content of both swallows, oat and salad, was more or less the same, as shown in Figure 29. This must be investigated one more time, using different features, time-based statistical features, or even using the same features if the microphone is changed.

Also when looking at the duration box plot, in Figure 27, it could be seen that the average duration of the oat bolus swallows is longer than salad bolus swallows. This could be exploited using a classifier built to distinguish between the food types using durations, as on average, dry food takes longer to swallow than moist food. This was not tested in this project, as the classifiers built performed better for longer segment lengths, thus duration information was harder to perceive. A shorter segment length should be used if the duration is to be used for such a purpose, as then there will be a larger difference in the amount of detected consecutive segments classified for swallows of different food types.

7.7 Final system design

The final design of such as system should also be looked into if the work done in this project is to be used in future research. Where and how should the microphones be placed should be investigated, as placing the microphone just above the collar bone proved to be problematic with regard to internal body noises. Also, placing it in that region was uncomfortable in the long run, with regard to head movement.

8 Bibliography

[1] D. Goldenberg and M. Hennessy, “Surgical anatomy and physiology of swallowing,” Apr 2016.

[2] Dansbecker, “Rectified linear units (relu) in deep learning,” May 2018.

[3] G. V. Jose, “Useful plots to diagnose your neural network,” Oct 2019.

[4] R. Gençay and M. Qi, “Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging,” *Neural Networks, IEEE Transactions on*, vol. 12, pp. 726 – 734, 08 2001.

[5] S. Shah, “Convolutional neural network: An overview,” Mar 2022.

[6] E. Orellana, “Breakdown confusion matrix,” Oct 2020.

[7] A. Isifan, “Combining bowel and swallowing sounds for improved meal detection,” project report in TFE4595, Department of Information Technology, and Electrical Engineering, NTNU – Norwegian University of Science and Technology, Dec. 2021.

[8] “Kyllingsalat - 225g grønnamp;frisk,” 2022.

[9] “Supergrøt - kanelamp;puffet quinoa 54g bare bra,” 2022.

[10] A. Loke, “Diabetes,” 2021.

[11] “What is diabetes?,” Dec 2021.

[12] M. clinic staff, “Hyperglycemia in diabetes,” 2020.

[13] NHS, “Gestational diabetes,” 2019.

[14] M. Dansinger, “Diabetes insulin pump: How it works,” 2021.

[15] “Insulin pumps & amp; continuous glucose monitors,” Mar 2022.

[16] Steven J. Russell, MD, PhD, Harvard Medical School, “Continuous glucose monitoring,” 2017.

[17] K. Kölle, “Feasibility of early meal detection based on abdominal sound,” Master’s thesis, Norwegian University of Science and Technology, 2019.

[18] M. Usman and H. Chen, “Recent trends in food intake monitoring using wearable sensors,” 01 2021.

[19] N. Alshurafa, H. Kalantarian, M. Pourhomayoun, J. Liu, S. Sarin, B. Shahbazi, and M. Sarrafzadeh, “Recognition of nutrition intake using time-frequency decomposition in a wearable necklace using a piezoelectric sensor,” *IEEE Sensors Journal*, vol. 15, pp. 1–1, 07 2015.

[20] J. Lee, P. Paudyal, A. Banerjee, and K. Sandeep, “A user-adaptive modeling for eating action identification from wristband time series,” *ACM Transactions on Interactive Intelligent Systems*, vol. 9, pp. 1–35, 10 2019.

[21] S. Zhang, R. Alharbi, M. Nicholson, and N. Alshurafa, “When generalized eating detection machine learning models fail in the field,” 09 2017.

[22] K. Kyritsis, C. Diou, and A. Delopoulos, “End-to-end learning for measuring in-meal eating behavior from a smartwatch,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5511–5514, 2018.

[23] S. Wang, G. Zhou, L. Hu, Z. Chen, and Y. Chen, “Care: chewing activity recognition using noninvasive single axis accelerometer,” 09 2015.

[24] C. Maramis, V. Kilintzis, and N. Maglaveras, “Real-time bite detection from smartwatch orientation sensor data,” 05 2016.

[25] O. Makeyev, P. Lopez-Meyer, S. Schuckers, W. Besio, and E. Sazonov, “Automatic food intake detection based on swallowing sounds,” *Biomedical signal processing and control*, vol. 7, pp. 649–656, 11 2012.

[26] H. Khlaifi, A. Badii, D. Istrate, and J. Demongeot, “Automatic detection and recognition of swallowing sounds,” 01 2019.

[27] T. Olubanjo and M. Ghovanloo, “Tracheal activity recognition based on acoustic signals,” *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014*, vol. 2014, pp. 1436–9, 08 2014.

[28] Y. Bi, M. Lv, C. Song, W. Xu, N. Guan, and W. Yi, “Autodietary: A wearable acoustic sensor system for food intake recognition in daily life,” *IEEE Sensors Journal*, vol. 16, no. 3, pp. 806–816, 2016.

[29] H. Khlaifi, “Preliminary study for detection and classification of swallowing sounds,” *Bioengineering. Université de Technologie de Compiègne,* 2019.

[30] "", “Normal swallowing,” *Sydney Voice and Swallowing*, 2021.

[31] M. Taniwaki and K. Kohyama, “Fast fourier transform analysis of sounds made while swallowing various foods,” *The Journal of the Acoustical Society of America*, vol. 132, no. 4, pp. 2478–2482, 2012.

[32] F. Arvin, S. Doraisamy, and E. Safar Khorasani, “Frequency shifting approach towards textual transcription of heartbeat sounds,” *Biological procedures online*, vol. 13, p. 7, 10 2011.

[33] D. F, A. MT, M. P, and Menier, “Acoustic properties of the normal chest,” 1995.

[34] R. Triggs, “What you think you know about bit-depth is probably wrong,” 2021.

[35] D. Greene, *Decimation and Downsampling*. Rice, 2021.

[36] U. Jaitley, “Why data normalization is necessary for machine learning models,” 10 2018.

[37] S. D. Yongxin Luo, “Power spectral density,” *Science Direct*, 2007.

[38] P. Nair, “The dummy’s guide to mfcc,” Jul 2018.

[39] D. Gartzman, “Getting to know the mel spectrogram,” May 2020.

[40] J. Lyons, “Mel frequency cepstral coefficient (mfcc) tutorial,” 2012.

[41] J. Hui, “Speech recognition-feature extraction mfcc amp; plp,” Sep 2019.

[42] D. Raj, “A note on mfccs and delta features,” Jul 2019.

[43] I. C. Education, “What is machine learning?,” Jul 2020.

[44] D. Stansbury, “A gentle introduction to artificial neural networks,” Jul 2020.

[45] I. C. Education, “What are neural networks?,” Aug 2020.

[46] J. Nduati, “Introduction to neural networks,” Oct 2020.

[47] A. S. V, “Understanding activation functions in neural networks,” 2017.

[48] B. K, “Understanding relu: The most popular activation function in 5 minutes!,” Nov 2020.

[49] R. Pramoditha, “How to choose the right activation function for neural networks,” Jan 2022.

[50] S. Saxena, “Softmax: What is softmax activation function: Introduction to softmax,” 4 2021.

[51] T. A. Myrvoll, “Deep learning deep neural networks,” Mar 2022.

[52] J. Brownlee, “Code adam optimization algorithm from scratch,” Oct 2021.

[53] J. Jordan, “Setting the learning rate of your neural network.,” Aug 2020.

[54] B. I. C. Education, “What is overfitting?,” 2021.

[55] J. Brownlee, “How to perform feature selection with numerical input data,” Aug 2020.

[56] G. Boesch, “Deep neural network: The 3 popular types (mlp, cnn and rnn),” Oct 2021.

[57] S. Saha, “A comprehensive guide to convolutional neural networks-the eli5 way,” Dec 2018.

[58] J. Jordan, “Evaluating a machine learning model.,” Aug 2018.

[59] K. Kölle, “Protocol for the pilot study: Analysis of bowel sounds related to meal onset,” 2021.

[60] Octopart, *SPM0687LR5H-1*, 2018.

[61] Roland, *Roland UA-1010 Octa-Capture*, 2020.

[62] “Calculating insulin dose,” 2022.

A Zip file

A zip file is also included with this thesis. This file contains most of the relevant code used in this project in addition to a couple of papers. The file contains the code for decimation, feature extraction (Mel spectrogram and PSD features), and the code for the four built classifiers, as shown in Figure 38. In addition, the zip file contains also Konstanze K. [17], and the previous project [7] paper, "Konstanze's paper" and "ProsjektOppgave".

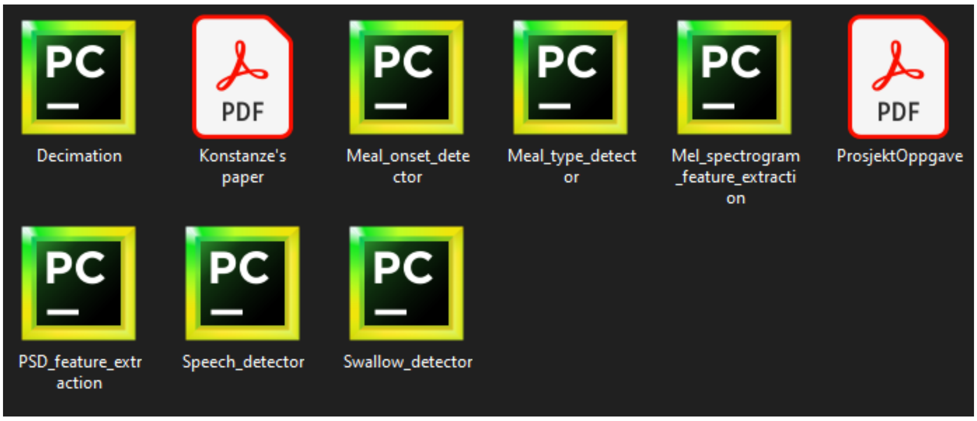


Figure 38: Included zip file content.

NTNU

Norwegian University of
Science and Technology