Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Informatyki

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Sztuczna Inteligencja

Application for detection and analysis of bowel sounds

Jakub Ficek
Numer albumu 293091

promotor
dr hab. inż. Robert Marek Nowak, prof. PW

WARSZAWA 2023

Application for detection and analysis of bowel sounds

Abstract. Bowel sound auscultation is a non-invasive measurement of intestine activity. It can be used for the diagnosis of diseases like irritable bowel syndrome. In contrast to heart and lung auscultation, bowel sound auscultation is not commonly used by doctors. The main reason is the expensive analysis of long recordings of bowel sounds. In this thesis, I developed the algorithm for the automatic detection and analysis of bowel sounds using convolutional and recurrent neural networks using sequences of spectrograms as input. I compared the acquired results with the results acquired using the sliding window approach and machine learning methods including logistic regression, support vector machine, random forest, and gradient boosting. I used data augmentation and semi-supervised learning methods to improve the developed neural network. The final model achieved 98.14% accuracy and 89.34% F1 score using the real data. I built the application for automatic generation of reports of bowel sound analysis and shared it on the website to be used by the doctors.

Keywords: bowel sounds, sound detection, machine learning, neural networks, data augmentation, semi-supervised learning

Aplikacja do detekcji i analizy dźwięków jelitowych

Streszczenie. Osłuchiwanie dźwięków jelitowych jest nieinwazyjną metodą pomiaru aktywności jelit. Może być stosowane do diagnozowania chorób, takich jak zespół jelita drażliwego. W przeciwieństwie do osłuchiwania serca i płuc, osłuchiwanie dźwięków jelitowych nie jest powszechnie stosowane przez lekarzy. Głównym powodem jest kosztowna analiza długich nagrań dźwięków jelitowych. W ramach tej pracy, opracowałem algorytm do automatycznej detekcji i analizy dźwięków jelitowych przy pomocy splotowych i rekurencyjnych sieci neuronowych używając sekwencji spektrogramów jako wejścia. Porównałem uzyskane wyniki z wynikami uzyskanymi przy pomocy metody przesuwnego okna i metod uczenia maszynowego włączając regresję logistyczną, maszynę wektorów nośnych, las losowy i wzmocnienie gradientowe. Użyłem metod augmentacji danych i uczenia częściowo nadzorowanego do polepszenia stworzonego modelu. Ostateczny model uzyskał 98.14% dokładności i 89.34% miary F1 na danych rzeczywistych. Zbudowałem aplikację do automatycznego generowania raportów analizy dźwięków jelitowych i udostępniłem ją na stronie do używania przez lekarzy.

Słowa kluczowe: dźwięki jelitowe, detekcja dźwięków, uczenie maszynowe, sieci neuronowe, augmentacja danych, uczenie częściowo nadzorowane

załącznik nr 10 do zarządzenia
nr 46 /2016 Rektora PW

Politechnika Warszawska
Warsaw University of Technology

……….......................
miejscowość i data
place and date

……………………………..
imię i nazwisko studenta
name and surname of the student

……………………………..
numer albumu
student record book number

……………………….……….
kierunek studiów
field of study

OŚWIADCZENIE

DECLARATION

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.
Under the penalty of perjury, I hereby certify that I wrote my diploma thesis on my own, under the guidance of the thesis supervisor.

Jednocześnie oświadczam, że:
I also declare that:

– niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
– *this diploma thesis does not constitute infringement of copyright following the act of 4 February 1994 on copyright and related rights (Journal of Acts of 2006 no. 90, item 631 with further amendments) or personal rights protected under the civil law,*

– niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
– *the diploma thesis does not contain data or information acquired in an illegal way,*

– niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
– *the diploma thesis has never been the basis of any other official proceedings leading to the award of diplomas or professional degrees,*

– wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
– *all information included in the diploma thesis, derived from printed and electronic sources, has been documented with relevant references in the literature section,*

– znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.
– *I am aware of the regulations at Warsaw University of Technology on management of copyright and related rights, industrial property rights and commercialisation.*

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.
I certify that the content of the printed version of the diploma thesis, the content of the electronic version of the diploma thesis (on a CD) and the content of the diploma thesis in the Archive of Diploma Theses (APD module) of the USOS system are identical.

....
czytelny podpis studenta
legible signature of the student

Contents

1. Introduction

1.1. Thesis goal and objectives

Bowel sound auscultation is a useful measurement of intestine activity. Especially, it can be used for noninvasive diagnosis of irritable bowel syndrome that affects 10% to 15% of the population. Besides many potential applications, bowel sound auscultation is not commonly used by doctors. The main reason is the need for long measurement time and very time-consuming analysis of recordings by an expert, so no research based on a big population was performed. For this reason, a tool for automatic detection and analysis of bowel sounds is needed.

The purpose of this thesis is to develop and share application for the analysis of bowel sounds. Another purpose is to improve the model proposed in my engineering thesis [1] using a bigger dataset, data augmentation, and semi-supervised learning methods and to compare it with other machine learning methods.

My research hypothesis is that the proposed bowel sound detection model can be improved using data augmentation and semi-supervised learning methods. An additional hypothesis is that the sliding window based methods can achieve comparable results to the sequence processing approach.

1.2. Thesis structure

This thesis was divided into 5 chapters. In Chapter 2 I described the existing methods of bowel sound analysis and its applications and I discussed the signal processing, machine learning, and neural network methods necessary to understand further work. In Chapter 3 I proposed the solution for bowel sound detection and described the implementation of the application and the bowel sound statistics. In Chapter 4 I proposed new methods and improvement techniques for the algorithm. I also analyzed the performed experiments and validated the quality of the developed models. In chapter 5 I summarized the thesis and pointed out potential work in the future.

2. Bowel sound processing

2.1. Bowel sounds characteristics

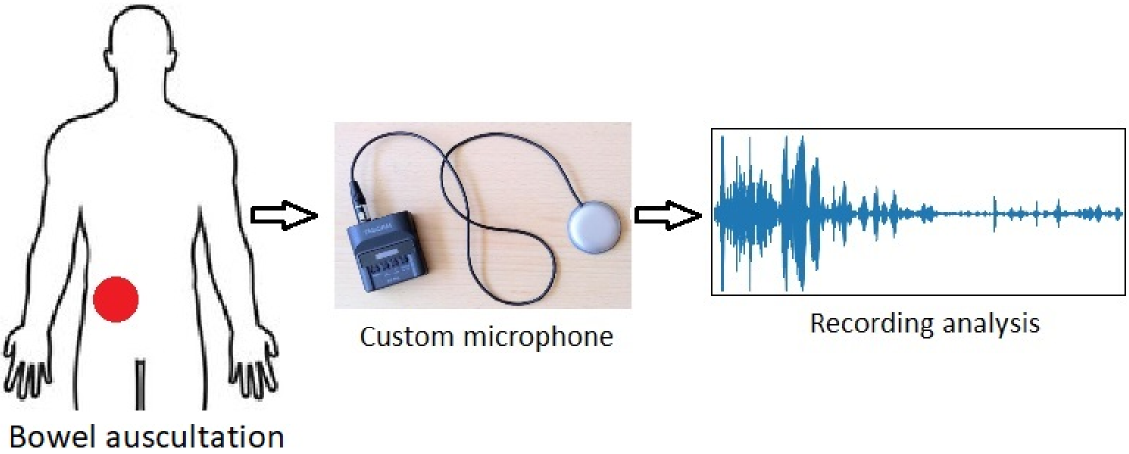


Figure 2.1. Procedure of the bowel sound analysis. The red circle shows the right lower abdominal quadrant where the microphone is put.

Bowel sounds are sounds generated by contractions of the intestine and allow doctors to measure the motoric activity of the digestion system. The scheme of the procedure is shown in figure 2.1. Figure 2.2 presents 4 types of bowel sounds that can be highlighted: single burst bowel sounds, distinct burst bowel sounds, multiple burst bowel sounds, and continuous random bowel sounds. In spectrogram, we can also see the noise below 200Hz that comes from heart beating and venous hum. Single burst bowel sounds are weak and make up approximately 85% of all sounds. They occur several times a second. Their duration is usually 10-40ms and frequency varies between 60Hz and 2kHz. Distinct burst sounds are louder and more visible in the spectrogram than single burst sounds and make up approximately 5-10% of all sounds. Their duration is similar to single burst sounds and their frequency can be up to 3kHz. Multiple burst sounds are single burst and distinct burst sounds that occur multiple times in a short period. They make up approximately 5% of all sounds and their duration can be even 1.5s. The rarest sounds are continuous random bowel sounds that make up approximately 1% of bowel sounds. Their duration can be even several seconds and are very irregular. They are associated with stomach rumbling that humans can hear.

This work focuses on detecting single burst bowel sounds because they seem to be the most useful type for quantitative research. The problem with the reliable detection of bowel sounds is noise occurring during long measurement. The main sources of noise are heartbeat, respiration sound, clothes friction, and noise coming from the environment.

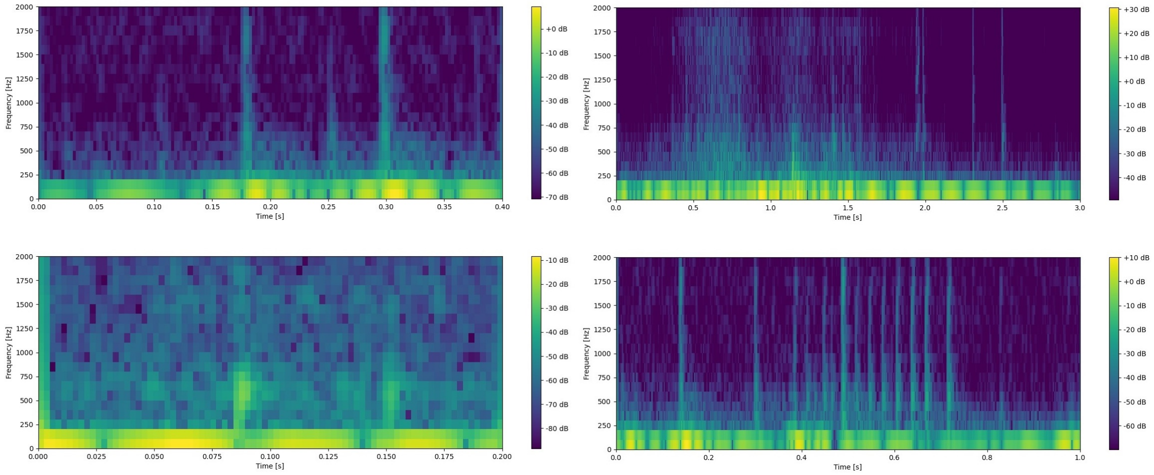


Figure 2.2. Spectrograms of different types of bowel sounds. The top left image shows two distinct burst bowel sounds in 0.18s and 0.30s. The top right image shows continuous random bowel sound from 0.4s to 1.6s. The bottom left image shows single burst bowel sounds in 0.085s and 0.150s. The bottom right image shows multiple burst sound from 0.3s to 0.7s.

2.2. Signal processing

Raw audio recording consists of amplitudes of sound in successive time intervals associated with a sampling frequency of the recording device. Sounds in nature are the combinations of waves, so the audio data is smooth and it is usually useful to convert audio to the basis of complex exponential functions. Discrete Fourier transform (DFT) is a linear operation that converts time domain signal $x = (x_0, ..., x_{N-1})$ to frequency domain signal $(X_0, ..., X_{\lfloor \frac{N}{2} \rfloor})$. Coefficients for frequency domain are defined by formula (1). A naive implementation of this algorithm has the $O(N^2)$ complexity because it requires multiplying a signal vector by a DFT matrix of size $N \times N$. Fast Fourier transform (FFT) algorithm achieves $O(NlogN)$ complexity by factorizing the DFT matrix.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i \frac{kn}{N}}, \quad k \in \{0, ..., N-1\} \tag{1}$$

$$f_k = \frac{k f_s}{N} \tag{2}$$

The frequency f_k associated with X_k coefficient is given by the formula (2), where f_s is a sampling frequency. Only first $\lfloor \frac{N}{2} \rfloor$ coefficients are used, so maximum frequency is $\frac{f_s}{2}$ according to the sampling theorem.

To get a two-dimensional spectrogram, a short-time Fourier transform (STFT) is used, defined by the formula (3), where $X_{k,m}$ represents the kth row and mth column of the spectrogram. The algorithm divides the time domain signal into FFT windows of size N and applies DFT for each window. The number of windows M depends on the FFT window size and window hop length. Because each window is assumed to be part of the periodic function, the window smoothing function w_n is used to multiply the input signal and smooth it on the edges and avoid high-frequency leakages. The window smoothing function used in this work is Hann window (4), which completely resets values on the edges.

$$X_{k,m} = \sum_{n=0}^{N-1} w_n x_{mN+n} e^{-2\pi i \frac{kn}{N}}, \quad k \in \{0,...,N-1\} \wedge m \in \{0,...,M-1\} \tag{3}$$

$$w_n = 0,5\left(1 - cos\left(\frac{2\pi n}{N-1}\right)\right) \tag{4}$$

The received $X_{k,m}$ is a complex value, so to get power spectrogram values, $|X_{k,m}|^2$ values are used. In this work, spectrogram values are additionally converted to logarithmic decibel scale to better reflect volume differences.

Mel-scale spectrograms are often used for audio analysis, which is the frequency scale that focuses more on low frequencies and better reflects differences in frequency heard by humans. It isn't used in this work, because the high-frequency part of the spectrograms are cut off, so it isn't necessary.

2.3. Machine learning

The machine learning model is the model that approximates some distribution based on the training data sampled from this distribution. The model contains parameters that are adjusted to minimize the certain loss function. The model needs to contain enough parameters to be able to properly fit the complex distribution, but the more parameters model contains, the more training data is required to adjust parameters properly. If the model is too complex for the training data size, the model fits the training data well but generalizes badly for other data, which is called model overfitting. This rule is called the bias-variance tradeoff.

Logistic regression is an example of the machine learning algorithm. It uses the logistic function on linear regression to produce a value between 0 and 1, so the input data for which the model produces a given value is a hyperplane. The model is described by the formula (5), where x is an input vector, w is a parameters vector and b is offset. The model is simple, so requires a small amount of training data. However, logistic regression is unable to model nonlinear functions.

$$p(x) = \frac{e^{wx+b}}{1 + e^{wx+b}} \tag{5}$$

Support vector machine (SVM) is another machine learning algorithm. The model also adjusts the hyperplane to make binary predictions. In contrast to logistic regression, SVM converts the data to higher dimensional space where the data can be linearly separable. To avoid the big complexity of the higher dimensional space conversion the kernel trick is used. The kernel trick is that the algorithm doesn't need to calculate the parameters of the data in the embedded space to calculate the dot product of two samples and optimize the model. Additionally, if the kernel function meets certain conditions, the optimization is convex. An example kernel is the radial basis function described by the formula (6), where x_1, x_2 are data samples.

$$K(x_1, x_2) = e^{-\frac{||x_1 - x_2||^2}{2\sigma^2}} \tag{6}$$

Random forest model, the next machine learning algorithm, uses a model ensemble approach. The method creates a set of decision trees to make the prediction. The motivation is that decision trees tend to overfit the data, so multiple decision trees are built by sampling the training data. Then, during inference, the majority of the outputs from all trees are used for the final prediction to reduce the variance of individual trees.

Gradient boosting model is another machine learning algorithm. It also uses a model ensemble approach. Instead of building all models simultaneously, the weak models are created in sequence, so the next model reduces the error of the previous models. Then, during inference, the sum of the outputs of all models is used for the prediction. The motivation is that weak models have low variance and the next models reduce the bias of the previous ones. The disadvantage of this method is that due to the dependencies of the weak models, parallelization is much more difficult compared to the random forest.

2.4. Neural networks

Neural networks are a family of machine learning models inspired by brain structure. Neural network models achieve the best results in complex tasks like natural language processing and computer vision. Currently, neural networks could have billions of parameters, therefore training is complex and requires a huge training dataset.

The simplest neural network model is a multilayer perceptron. It contains an input layer, several hidden layers, and an output layer. Each layer consists of a certain number of neurons. The neurons of the given layer are fully connected with the neurons of the next layer. It means that the outputs of the given layer are passed as an input to the next layer. The neuron output is described by the formula (7), where x_i is the ith input of the neuron, w_i is the ith weight of the neuron, and φ is the activation function that enables modeling nonlinear relations. The most common activation function is ReLU (8). For the classification neural network, to produce the final output, the sigmoid function (9) is used.

$$y = \varphi\left(\sum_{i=0}^{N} w_i x_i + b\right) \tag{7}$$

$$ReLU(x) = max(0, x) \tag{8}$$

$$\sigma(x) = \frac{e^x}{e^x + 1} \tag{9}$$

The most common loss function for neural network classifiers is binary cross-entropy given by formula (10).

$$Loss = -\frac{1}{N}\sum_{i=0}^{N}\left(y_i log(\tilde{y_i}) + (1-y_i) log(1-\tilde{y_i})\right) \tag{10}$$

The weights of neurons are updated to minimize the loss function by calculating the gradients of weights. To make calculating the gradients of the parameters more effective, a backpropagation algorithm is used. It calculates the gradients of the nth layer using previously calculated gradients of the (n+1)th layer by using the derivative chain rule described by formula (11).

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx} \tag{11}$$

The number of multilayer perceptron parameters grows fast with increasing the dimension of input due to the fully connected approach. To avoid it, the convolutional neural networks (CNN) are used. In the convolutional layer, instead of the fully connected layers, neurons are connected locally sharing the same weights by using convolution operation. It is equivalent to using filters with adjustable parameters. Outputs from applying many filters are the activation maps that can be passed to the next convolutional layer. Convolutional neural networks are especially used in computer vision, because using filters, makes the model less prone to data translation.

Recurrent neural network (RNN) is a model designed to handle sequences of data. RNN holds an additional hidden state, that gets updated when processing the input sequence. This way, the sequence element is processed based on the input from the element and the hidden state produced from the previous element which is summarized in formula (12, where x_n is the nth element of the sequence, h_t is the hidden state produced from the nth element and W is the weights matrix.

$$h_n = \varphi(Wx_n + Uh_{n-1}) \tag{12}$$

The disadvantage of this model is that the backpropagation algorithm needs to process back the whole sequence to calculate the gradients. Due to the long backpropagation chain, the problem of unstable gradient occurs and the gradients become very small or very large. To handle this problem, Long short-term memory (LSTM) and gated recurrent unit (GRU) [2] were created. These methods introduce memory gates that allow neural networks to focus more on relevant parts of the sequence. GRU method is summarized in formula (13), where z_n is the update gate, r_n is the reset gate and W, W_z, W_u are the standard weight matrix, weight matrix of update gate, weight matrix of reset gate respectively.

$$\begin{aligned}
h_n &= (1 - z_n)h_{n-1} + z_n\tilde{h_n} \\
z_n &= \sigma(W_z x_n + U_z h_{n-1}) \\
\tilde{h_n} &= tanh(W x_n + U(r_n \otimes h_{n-1})) \\
r_n &= \sigma(W_r x_n + U_r h_{n-1})
\end{aligned} \tag{13}$$

The motivation is that the reset gate reduces the impact of the state of not relevant sequence element and the update gate controls the rate of the new state update. The LSTM approach is similar but introduces a third gate, so it has more parameters.

The disadvantage of RNN models is that due to recurrent flow, the model training isn't parallelizable, so the training is time consuming, especially for long sequences.

2.5. Regularization

Regularization is constraining the model parameters in some way. The motivation is to reduce the variance of the model and make it less likely to overfit. The easiest method is to constrain the parameters to small values by adding weighted L2 loss (14) to the loss function.

$$L2 = \sum_{i=0}^{N} w_i^2 \tag{14}$$

Another approach, applicable only to neural networks is dropout [3]. It is the method that involves the random removal of part of neurons during model training and the use of all neurons during inference. The motivation is to force the neural network to use different subsets of neurons and not rely only on one subset. Using fewer neurons during training reduces the variance of the model similar to the model ensemble approach.

Data augmentation is a form of regularization that involves using additional data during training by slightly modifying the possessed data. The motivation is that it helps to reduce overfitting because small modifications of data shouldn't change the output of the model. It is especially useful for computer vision tasks because images are easy to augment without affecting labels by transformations like rotation or shifting.

One of the data augmentation techniques that can be used independently of the domain is MixUp [4]. It involves mixing two samples from the training data and creating a new one by interpolating the features and the labels based on λ parameter sampled from $\beta(\alpha, \alpha)$ distribution (15). The bigger the α value is, the samples are interpolated more equally.

$$\begin{aligned}
x_{new} &= \lambda x_i + (1 - \lambda)x_j \\
y_{new} &= \lambda y_i + (1 - \lambda)y_j \\
\lambda &\sim \beta(\alpha, \alpha)
\end{aligned} \tag{15}$$

The motivation of MixUp is to increase the robustness of the model and to make it less prone to corrupt labels.

2.6. Semi-supervised learning

Semi-supervised learning (SSL) is the method that involves using labeled and unlabeled data to train the model. It is usually used when a small amount of labeled data and a big amount of unlabeled data is available. Semi-supervised learning can improve the model because real data feature space should be smooth and similar samples should have similar labels in most cases. Two main semi-supervised methods are pseudo-labeling and consistency regularization.

Pseudo-labeling involves labeling unlabeled samples using the probabilities given by a model trained on labeled samples. Then, a new model is trained using additional pseudo-labeled samples. The motivation is that using hard labels from model probabilities improves separating data into clusters by the new model which is called entropy minimization. In this method, the model that creates pseudo-labels is called the teacher and the model trained using pseudo-labels is called the student.

Consistency regularization involves using data augmentation on unlabeled samples. Training involves learning the model to predict the same labels on augmented and not augmented samples. It benefits the model similarly to data augmentation in supervised learning but doesn't need prior labels.

In this work, two semi-supervised learning methods are tested: Curriculum Labeling and FixMatch described in detail in Chapter 4.

2.7. Machine learning model quality metrics

After developing the machine learning model, the method for determining its quality is needed. It is done by calculating various metrics based on the ground truth and predictions of the model on the test dataset that is independent of the training data.

Sometimes, to lower the variance of metrics, the cross-validation method is used. It involves dividing the whole dataset into n parts and training the model n times using a different part as a test dataset each time.

We can distinguish four results of binary prediction:

- true positive (*TP*) - correct prediction of presence
- true negative (*TN*) - correct prediction of lack of presence
- false positive (*FP*) - incorrect prediction of presence
- false negative (*FN*) - incorrect prediction of lack of presence

When dealing with an imbalanced dataset, accuracy (16) can be misleading (high accuracy is achieved by the model that always predicts the majority class). For this reason, based on the four results mentioned above, precision (17), recall (18), specificity (19), and F1 (20) are used.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$Spec = \frac{TN}{TN + FP} \tag{19}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{20}$$

In the case of models generating continuous output, the metrics depend on the chosen probability threshold. To get a more general score, the ROC AUC metric is used. It is defined as the area under the curve plotted from the true positive rate (recall) and false positive rate (1-specificity) for different threshold values. Similarly, PR AUC is defined as the area under the curve plotted from the precision and recall for different threshold values.

2.8. Quality of bowel sound analysis

The metrics described in the previous section can be used in bowel sound detection task if we divide the recording into frames representing small time intervals and classify each frame as a bowel sound interval or silent interval. The disadvantage of metrics calculated this way is that they don't directly inform about the success rate and accuracy of predicting the whole bowel sounds. Another disadvantage is that using metrics based on frame classification can't be directly calculated and compared for methods that use different frame size or don't use frame division.

The main purpose of bowel sound analysis is to measure the frequency and duration distribution of bowel sounds, so it is important to also measure metrics based on bowel sounds instead of frames. For this reason, I calculated the IOU value that is widely used in object detection tasks in computer vision.

$$IOU(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{21}$$

The formula is given in (21). It calculates the ratio of the intersection of the ground truth and predicted time interval to the union of the ground truth and predicted time interval. If the IOU value is above a certain threshold, it is treated as TP. If there is no IOU value above a certain threshold for a given ground truth sound, it is treated as FN. If there is no IOU value above a certain threshold for a given predicted sound, it is treated as FP. Based on these results, we can calculate IOU precision, IOU recall, and IOU F1. We don't calculate IOU specificity, because there is no TN defined. The example of the union and intersection of intervals is shown in figure 2.3.

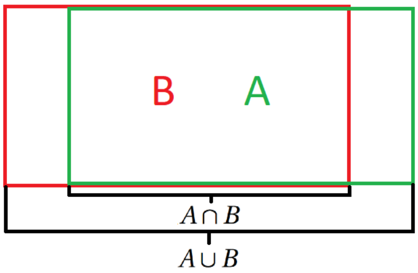


Figure 2.3. Scheme of union and intersection of A and B intervals. A is represented by a green rectangle and B is represented by a red rectangle.

2.9. Literature overview

Research on intestinal sounds has been going on for several decades with many developed methods and use cases. The approaches for automatic detection of bowel sounds can be divided into 3 main categories that are often combined: using filters, using machine learning models, and using deep learning methods.

The quantitative analysis of publications related to bowel sound analysis methods is shown in figure 2.4. We can see that the number of publications in this domain is consistently growing. The older publications contain mostly filtering and machine learning methods. In recent years, we can see a big increase in publications introducing deep learning methods. Especially in the years 2020-2022, over half of the publications are related to deep learning models. In modern approaches, the filtering and machine learning methods occur less frequently as a complete solution but are often added as a preprocessing step.

The big problem in comparing the accuracy of various detection models is the lack of a clear definition of bowel sounds. Additionally, there is no generally recognized bowel sound dataset. For this reason, direct comparing metrics achieved by different researchers presented in this subsection should be treated with caution.

I am co-author of an overview of automated bowel sound analysis, published in 2021 [5]. It summarized developed methods and their quality. The work noted that despite many prototypes, there are no publicly available tools for clinicians. The lack of uniform methodology, international forums for discussion, and publicly available benchmark datasets were mentioned as the main problems.

2.9.1. Method

On April 2023 Scopus was queried with the term: ("bowel sounds" OR "abdominal sounds" "neural network" OR "machine learning" OR "filter" OR "deep learning") and Google Scholar was queried with the term: ("bowel sounds" "analysis" "automation"

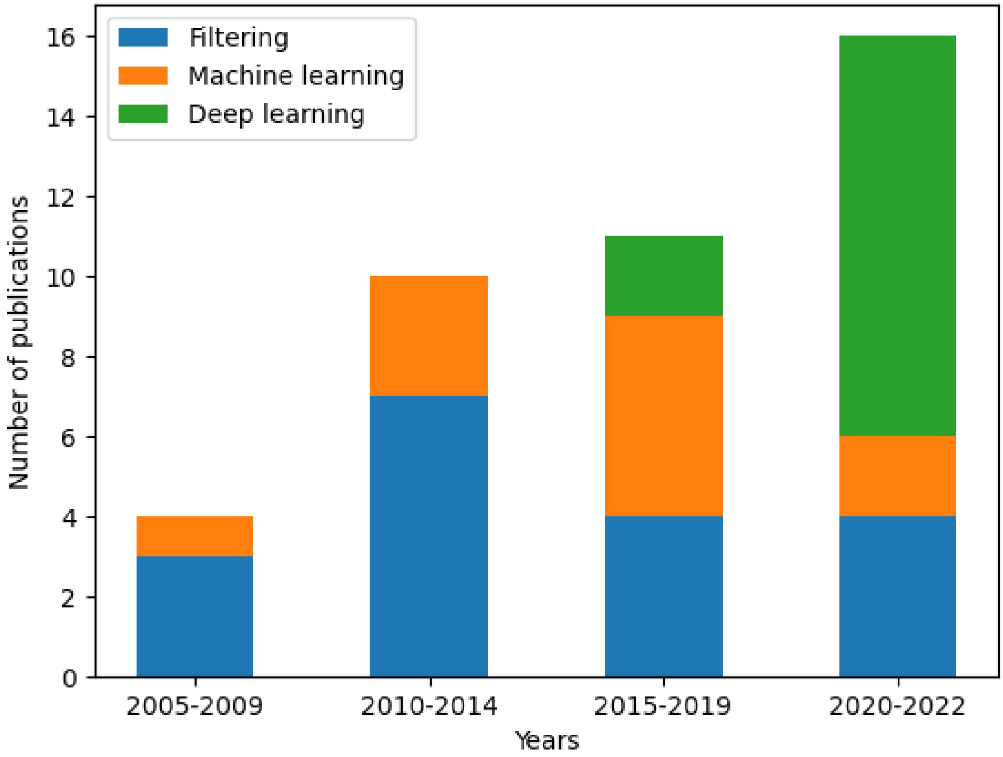


Figure 2.4. Number of publications related to automatic bowel sound analysis methods in years 2005-2022.

OR "algorithm"). The searches yielded 61 publications from Scopus and 39 from Google Scholar. 33 publications were chosen that cited 817 publications. From 817 cited publications, publications cited 2 times or more were selected and merged with the initially selected 33 publications, obtaining 64 publications. Finally, 53 publications were selected for analysis, based on the abstracts review. Publications before the year 2005 weren't included during the process except [6], [7], and [8].

2.9.2. Applications

One of the first studies of bowel sounds was published in 1955 [6]. The authors analyzed bowel sounds qualitatively and used bowel sounds frequency to classify and determine gastrointestinal motor activity.

Long-term bowel sound monitoring was applied to detect patterns related to digestion [9]. The authors found that bowel sound occurrence frequency was significantly higher after eating compared to fasting. It confirmed that bowel sound analysis is useful for controlling intestinal activity.

In another work [8], bowel sound intervals were analyzed for patients with Crohn's disease and irritable bowel syndrome (IBS). It turned out, that bowel sounds duration mean was significantly shorter for patients with IBS which didn't apply to healthy patients and patients with Crohn's disease.

Bowel sounds were used for diagnosis of ascites [7]. In this paper, a wavelet transform-based stationary-nonstationary filter was used to denoise the audio. Then, the denoised signal was used to measure various statistics. The bowel sounds of patients with ascites had different structures and could be identified by higher-order component analysis.

Bowel sound analysis was used to detect spinal cord injury [10] [11]. For the detection of bowel sounds, an iterative kurtosis-based detection algorithm (IKD) was used. that was specially modified for this task. Then, jitter and shimmer features extracted from segments of bowel sounds were used as an input to a simple multilayer perceptron (MLP) and regression model that predicted conventional colon transit time (CCT) [12]. Delayed CCT is one of the symptoms of spinal cord injury, so the system could successfully detect the disease based on 19 patients.

A system based on statistical features was developed for early detection of bowel activity after operation [13]. The method allowed for the live detection of bowel sounds based on a cutoff value in the frequency band of the power spectrum. I tested this method in my engineering thesis and achieved a low recall of 13.83%, but the purpose of this system was to detect the resumption of bowel function, so recall of individual bowel sounds wasn't that important. In later work [14], the authors replaced the cutoff value method with a naive Bayes model based on spectral features which achieved 94.15% accuracy.

Bowel sounds analysis was used for early meal detection for insulin dosing to control glucose level for diabetic patients [15]. The authors used SVM with radial basis kernel and features based on power in a given frequency range as an input. The method allows to detect meals in 10 minutes on average, which is substantially faster than standard methods based on glucose monitoring that need approximately 30 minutes. In another work [16], various early meal detection models were developed. The best results were obtained using a convolutional neural network which was able to detect meals in 3 minutes on average without false positives.

Bowel sounds analysis allowed for classifying hyperfunction and paralysis of the intestine [17]. Signal energy and adaptive filtering were used for the detection and filtering of bowel sounds. Then, features like square root amplitude and kurtosis value were used for classification.

Bowel sounds were used for stress level prediction [18]. Data were collected when patients were doing relaxing and stressful tasks. Then, SVM and multilayer perceptron models using audio-based features were used for predicting the stress level. The best model achieved 75% accuracy and 75% F1 score.

2.9.3. Filtering methods

The relative breakthrough in bowel sounds analysis was presented in [19], [20]. The authors developed the piezoelectric sensor for recording and analyzed and proposed categorization of bowel sounds into five groups: single burst, multiple bursts, continuous random sound, harmonic sound, and a combination sound. Additionally, they calculated various statistics for each type, including duration time, spectral bandwidth, and mean-cross ratio. The authors also described the detected migrating motor complex cycle phases based on the bowel sounds characteristics and distinguished stages of signal processing of bowel sounds summarized in figure 2.5. In further work, the authors proposed a mathematical formula [21] of bowel sound based on 4 parameters related to bowel dimensions, the pressure inside the bowel, and the muscle contraction and relaxation pattern that allow generating synthetic bowel sounds. The formula was modified and used to reconstruct the bowel sounds occurring in the existing recordings in [22]. The formula was able to correctly approximate the bowel sounds with relatively good accuracy.

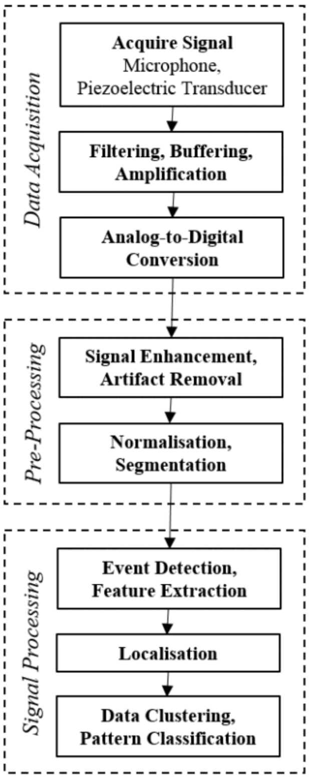


Figure 2.5. Stages of bowel sound signal processing from [19].

Wavelet transform with Wiener filtering was successfully used for denoising bowel sounds [23]. The denoised signal was used for automatic detection and segmentation [24]. Then, a multilayer perceptron using specially extracted features was used for the classification of bowel sounds and occurring noises [25]. All of this was used to develop an expert system *AIMAS* [26] which steps are summarized in figure 2.6. Despite building the complete system of bowel sound analysis, the downside is that this work focused mainly on more distinct bowel sounds, and single burst bowel sounds, which are the weakest and most difficult to detect, weren't taken into account.

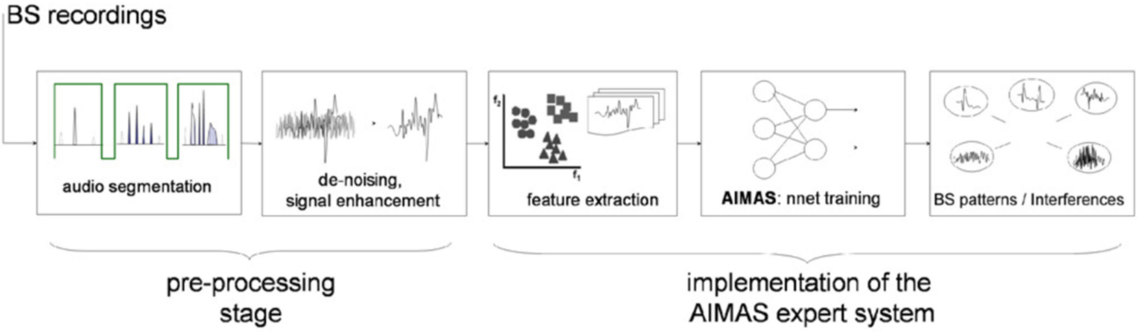


Figure 2.6. Stages of *AIMAS* expert system from [25].

In another work [27] a monitoring system in real-time was implemented in Labview for controlling the gastrointestinal motor activity after an operation. The authors used a bandpass filter and amplifier to make the bowel sound signal more visible to the physician.

Another approach for bowel sound detection was using the auto-regressive moving average model (ARMA) [28]. The model classified binary segments consisting of 256 time domain samples with 25% overlap. The authors declared a high sensitivity of 87.8±5.88% and specificity of 91.7±4.33% of the model.

Higher-Order-Statistics-Based fractal dimension algorithm, inspired by measuring the complexity of patterns in geometry, was used for bowel sounds detection [29]. In this work, audio was resampled to 2kHz and divided into 10ms frames. Additionally, the impact of injecting Gaussian noise and obtaining different signal-to-noise-ratio on algorithm accuracy was measured. The authors declared 77.27% accuracy and 82.25% recall.

Method for filtering bowel sounds was proposed in [30]. It involved decomposing the signal into intrinsic mode functions (IMF) using multivariate empirical mode decomposition (MEMD) that works similarly to a bandpass filter. Sliding windows of fractal dimensions were obtained from IMFs to denoise the signal. Then, based on heuristic rules related to power ratio, artifacts were eliminated from the signal to increase the precision of the method. The algorithm obtained 98.5% recall before artifact elimination and 76.1% after artifact elimination.

Bowel sound detection was successfully implemented using seasonal autoregressive integrated moving average (SARIMA) [31]. The bowel sound measurement system [32] was developed involving a microphone and vibration sensor that can be used for choosing the optimal time of medication. In further work, the authors developed the device customized for recording bowel sounds of moving patients [33]. For noise related to cloth friction that appears for moving patients, they implemented a noise cancellation algorithm using wavelet transform and rules based on bowel sound spectrum characteristics.

An adaptive line enhancer with the adaptive filter was used for enhancing the auscultation signal [34]. The work focused on denoising respiration and heart sounds as two main sources of auscultation noise.

2.9.4. Machine learning methods

Radial basis function (RBF) neural network was used for enhancing bowel sounds signal [35]. The idea was to use higher order statistics for suppressing noise with Gaussian and symmetrical distribution. The physicians confirmed that bowel sounds became more audible after applying this method.

Bowel sounds analysis was used for automatic recognition of digestion state [36]. Firstly, adaptive filters were used for denoising signal. Then, hand-engineered features from the time and frequency domain were extracted from the denoised signal and passed to a multilayer perceptron with 60 hidden neurons. 70% accuracy of predicting digestion state was obtained based on the experiments on 3 patients.

Legendre polynomials were used for bowel sound detection in [37]. The audio was divided into 1s frames and Legendre fitting was used for classification. In further work [38], SVM with radial-basis kernel was used and achieved 90.17% accuracy. Additionally, a monitoring system was developed that allowed for wireless recording and sending the data via Bluetooth in real-time. Due to the very big frame size of 1s, this work analyzed only very long bowel sounds.

Another use of multilayer perceptron for bowel sound detection was proposed in [39]. This work used power-normalized cepstral coefficients as an input to the model that obtained approximately 93% accuracy in a quiet environment and 90.6% in a loud environment. Based on the results, the impact of consumption on bowel sounds statistics was measured [40]. It turned out that the average frequency and duration significantly increased after the consumption of soda and coffee.

2.9.5. Deep learning methods

The first use of deep learning methods for bowel sound detection was described in [41]. The recording was divided into 100ms frames and classified using MFCCs. The recurrent neural network with LSTM layers was used. The model achieved 92.56% accuracy and 90.92% recall.

In 2019 skin-mounted wireless device for recording bowel sounds was developed [42]. The device was flexible to avoid deformation during changes in the patient's position and transmitted the signal to the computer in real life. The device had a fixed 2kHz sampling rate and was more sensitive to low frequency sounds than the standard stethoscope. For validation, a simple neural network for bowel sounds detection was developed that achieved 76.89% accuracy and the impact of food intake on bowel sounds was examined. In further work [43] in 2022, active noise reduction was built into the improved device. It relied on the second microphone that was recording the ambient noise only and was used for fine-tuning the adaptive filter used on the bowel sound microphone. Based on the improved device, the CNN model was developed [44]. The model achieved 91.06% accuracy. The frame size was set to 5ms, so the method was able to include the important single burst bowel sounds.

CRNN model was developed for the classification and detection of bowel sounds and noises [45]. The authors recorded 60 hours of audio data from 120 patients with different diseases and divided them into 5s fragments. The fragments were labeled into 6 categories: bowel sounds and different kinds of noises. The developed CRNN consisted of 5 CNN layers followed by a bidirectional GRU layer. Spectrograms, MFCC, and filter banks combined were used as input to the model. Two tasks were highlighted: predicting category for 5s segments and detecting sound occurrences based on 1s frames. The authors declared an 81.06% F1 score on the classification task and a 70.13% F1 score on the detection task. Although the achieved results are good, 1s frames are too long for reliable detection of short bowel sounds.

Another deep learning approach was published in 2020 [46] [47]. Multiple sensors system was developed and a convolutional neural network with spectrograms as input was used for the detection task. The model achieved 76.8% accuracy with 67% recall and 86.5% specificity. The advantage of this model was its relatively small size of 20.35k parameters, which allowed for the integration algorithm into the recording device. In further work [48], a U-Net neural network was proposed for the segmentation and enhancement of bowel sounds. The U-Net model allowed for increasing the signal-to-noise ratio of the recordings by 28.72dB on average.

CNN model with Laplace Hidden Semi-Markov Model (HSMM) was used for long bowel sound detection [49] of infants. This task is more difficult because neonatal bowel sounds are weaker and more noisy compared to adults. The dataset consisted of recordings from 49 infants divided into 6s segments. CNN with MFCC features was used to output the probability of segments. Then, the HSMM model was used to set time intervals based on output and duration probabilities that allow the method to improve accuracy on difficult segments. In further work [50], an ensemble of 2D CNN and 1D CNN models was used to improve the model. The developed method achieved 95.1% accuracy and 85.6% AUC.

The bowel sound characteristics were also analyzed for infants in [51]. The impact of breast milk intake, hyperbilirubinemia, and phototherapy on bowel sound parameters like frequency, duration, and amplitude was measured. The device with multiple sensors and CRNN model with MFCC features were used for the analysis system.

Recently, the method for bowel sound analysis using a built-in smartphone microphone was developed [52]. The authors created annotated dataset consisting of records from 100 patients using a consistent methodology with the one used on the dataset used by me. The CNN model using mel-scale spectrograms was built and achieved 89% accuracy and 72.3% F1 score. Additionally, they recreated the CRNN model based on my work and achieved a 66% F1 score on their dataset.

2.9.6. My method comparison to previous work

The method presented in this thesis involves using a single customized microphone for recording bowel sound during the night and classifying the frames of spectrograms using CRNN model and semi-supervised learning. Bowel sound analysis methods with the method proposed in this thesis are summarized in table 2.1.

Table 2.1. Summary of bowel sound analysis methods.

Filtering	Machine learning	Deep learning
MEMD ([30]) IKD ([10], [11]) Naive Bayes ([14]) SARIMA ([31]) ARMA ([28]) Fractal dimension ([29]) Wavelet transform ([7], [23], [33]) Adaptive filtering ([17], [34], [37])	SVM ([15], [30], [38]) MLP ([12], [18], [25], [36], [39]) RBF NN ([35]) Legendre fitting ([38])	CNN ([16], [44], [46], [52]) LSTM ([41]) CNN+HSMM ([49], [50]) U-Net ([48]) CRNN ([45], [51], [53]) **CRNN+SSL (this)**

In some of the previous work, multiple microphones were used to enhance the signal ([46], [47], [51]). In my method, only one microphone localized in the right lower abdominal quadrant is used. It is because the expert determined that a single microphone picks up sounds well enough and using more is not necessary.

In my method, the recordings are obtained from the patients during sleep. Recording in sleep allows for lower signal-to-noise-ratio, so the detection is easier but makes the method less flexible in comparison to methods that analyzed the bowel sound of moving patients ([33]).

In some of the previous work, the analysis algorithm was embedded into the device ([26], [33], [46], [47]). My method uses pre-recording but there aren't any constraints that disallow embedding the algorithm into the device in future work.

Some of the previous work uses signal enhancement and artifact removal techniques before model analysis ([7], [23], [35], [36]). My method doesn't involve these steps and uses only a low pass filter. The motivation is that complex machine learning models like deep neural networks should be able to solve the detection task using only slightly modified original signal.

My method uses deep learning models and frequency based signal representation like spectrogram for analysis, similar to the more recent work ([45], [46], [52], [51]).

The most distinctive part of my work is the use of data augmentation and semi-supervised learning that wasn’t tested previously. This approach is promising because it is widely used to improve machine learning models for medical domain tasks.

3. Application project and implementation

3.1. Solution overview

The scheme of the method proposed in my engineering thesis and also used in this work is shown in figure 3.1. Firstly, the recording is preprocessed. Because bowel sounds appear only on low frequencies, a low pass filter is used to cut off frequencies higher than 2kHz (increased from 1.5kHz in comparison to my engineering thesis) and a spectrogram is created from the time domain signal. Hop length is set to 2.5ms, FFT window size is set to 10 ms, and Hann window is used for the spectrogram creation. Spectrogram values are standardized based on the average and standard deviation obtained from the training dataset.

To detect bowel sounds, the recording is divided into 10ms frames. The frame is classified as a bowel sound if the bowel sound time interval includes more than half of the frame. The model makes binary classification on each frame using the generated probability value of bowel sound occurrence. After model predictions, we merge neighboring frames of bowel sounds to obtain time intervals and calculate statistics based on them.

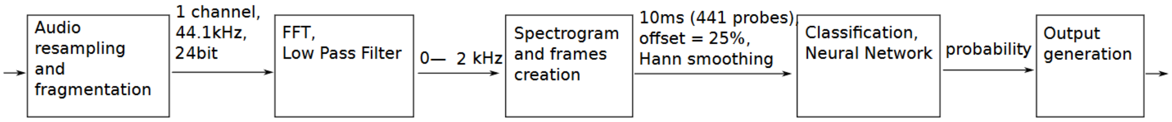

Audio resampling and fragmentation
1 channel, 44.1kHz, 24bit
FFT, Low Pass Filter
0— 2 kHz
Spectrogram and frames creation
10ms (441 probes) offset = 25%, Hann smoothing
Classification, Neural Network
probability
Output generation

Figure 3.1. Schema of the method adapted from [53].

3.2. Base model

The base approach is the sequence method tested in my engineering thesis. A sequence of frames is taken and used as an input to the model. Firstly, a features vector from each frame is created using convolutional layers. Then, these feature vectors are passed as input to the RNN layers. GRU layers are used, instead of LSTM, because they require fewer parameters and the long-term dependencies aren't so important in bowel sound analysis. An overview of the developed model is shown in figures 3.2.

This CRNN model is used as a base model for testing new approaches in this thesis. Compared to the application from my engineering thesis, the main extensions of this thesis are refactoring the code to make the Trainer class depend on Loader and Model interfaces to make it easier to add new approaches, implementing methods using the sliding window method, augmentation transformations, and semi-supervised learning, implementing RNN model, implementing IOU based metrics calculating, implementing reporting module to generate bowel sound analysis reports, and implementing a website and packaged application to use by doctors and researchers.

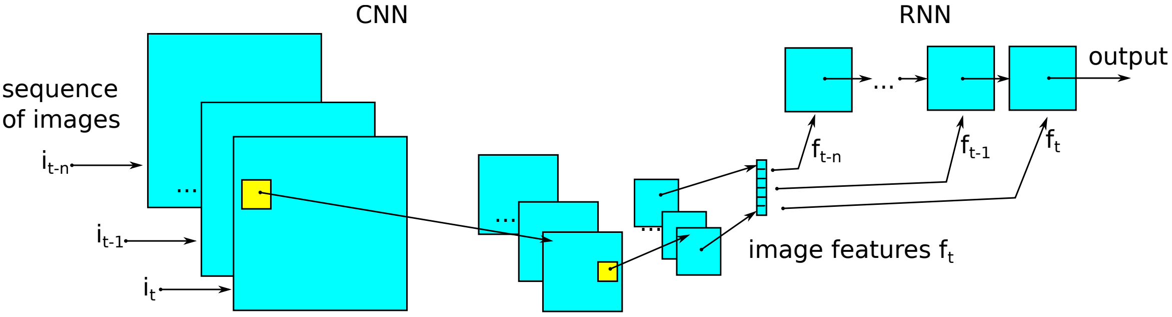


Figure 3.2. Architecture of the CRNN model coming from [53].

3.3. Modules description

The project is divided into the following modules:

- Preprocessing module splits the data into sets and unifies the format of the annotation files.
- Loading module loads data and converts it to spectrograms, it also applies augmentation transformations.
- Training module defines the structure of models, trains the models, and saves the results.
- Inference module makes predictions using trained models.
- Visualization module visualizes the ground truth and predicted bowel sounds on spectrograms.
- Reporting module creates reports containing bowel sound analysis based on the predictions acquired from inference.

The relation between the modules and the flow of the application are summarized in figure 3.3.

3.4. Data processing

The annotation files come into TXT from Audacity and CSV formats. Both file formats are unified into CSV files and incorrect labels (labels that contain bowel sounds of length 0) are removed during preprocessing. Then, based on the configuration file, the WAV files and associated annotations are split into a train, validation, and test directory.

Code for loading data into memory is divided into two classes inherited from the abstract class Loader: SequenceLoader and NeighborLoader. SequenceLoader is used for models using the sequence approach. The object loads samples, converts them into spectrograms, and divides spectrograms into frames. NeighborLoader is used for models using the sliding window approach. The object loads samples and converts them into spectrograms with overlapping contexts. Additionally, appropriate augmentation transformations are applied in this module according to the configuration file.

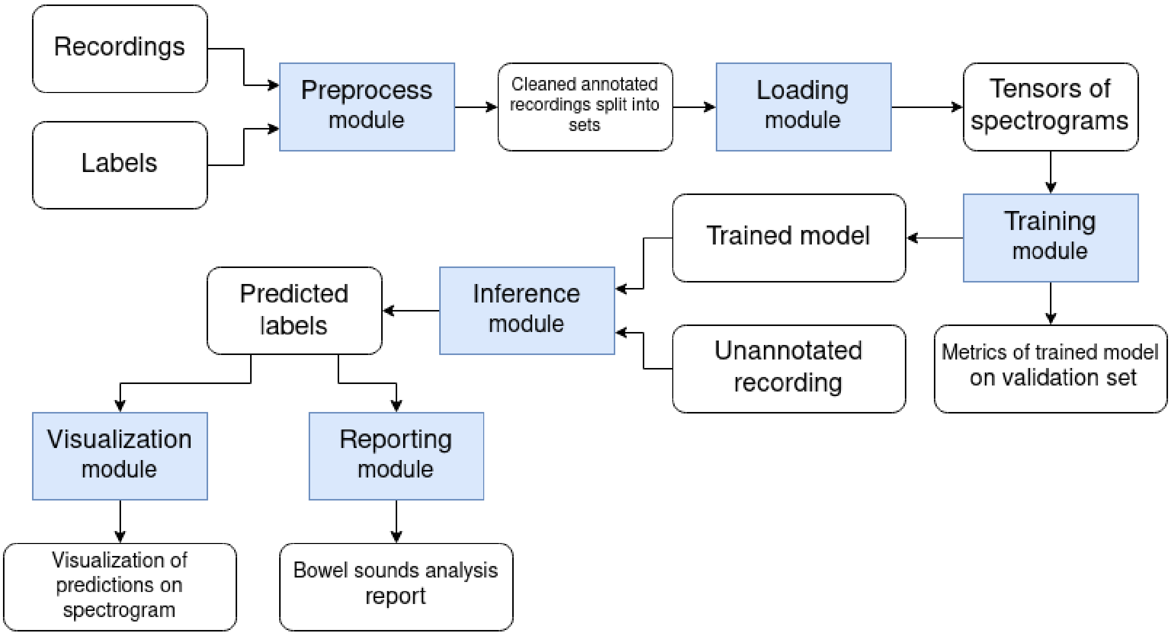


Figure 3.3. High-level flow of application and relations between the modules.

3.5. Model training

The main training process is implemented by the Trainer class. The Trainer object uses Model, Semisupervised, and Loader abstract classes to train the model and obtain metrics on the validation set after the training. The Loader object is used to load data into memory and pass it to the models. The Model class is used for training and is implemented by CRNN, RNN, LinearRegression, SVM, RandomForest, and GradientBoosting classes. Each one contains model architecture and training hyperparameters and implements methods for training, inferring, and saving the model. Semisupervised class is implemented by FixMatch and CurriculumLabel classes. Each one implements a training procedure for a given semi-supervised learning strategy. After the model is trained, three files are saved into the system: the H5 file containing the saved model, the TXT file containing the model metrics on train and validation sets, and the YML file containing parameters that allow to correctly process the data for model input. YML file also allows users to reliably recreate the experiment. Figure 3.4 shows a UML diagram of relations of classes that are used for training the model. For simplicity, only a subset of Model class implementations are shown in the diagram.

To manage experiments WandB tool is used. It allows users to visualize the learning curves and save the models and logs for each experiment in the cloud.

3.6. Inference and visualization

After training the model, the Inference class is used for obtaining the bowel sound time intervals on the recordings. It loads the audio signal in batches and converts them into spectrogram tensors. Then, inferring method of selected Model class implementation

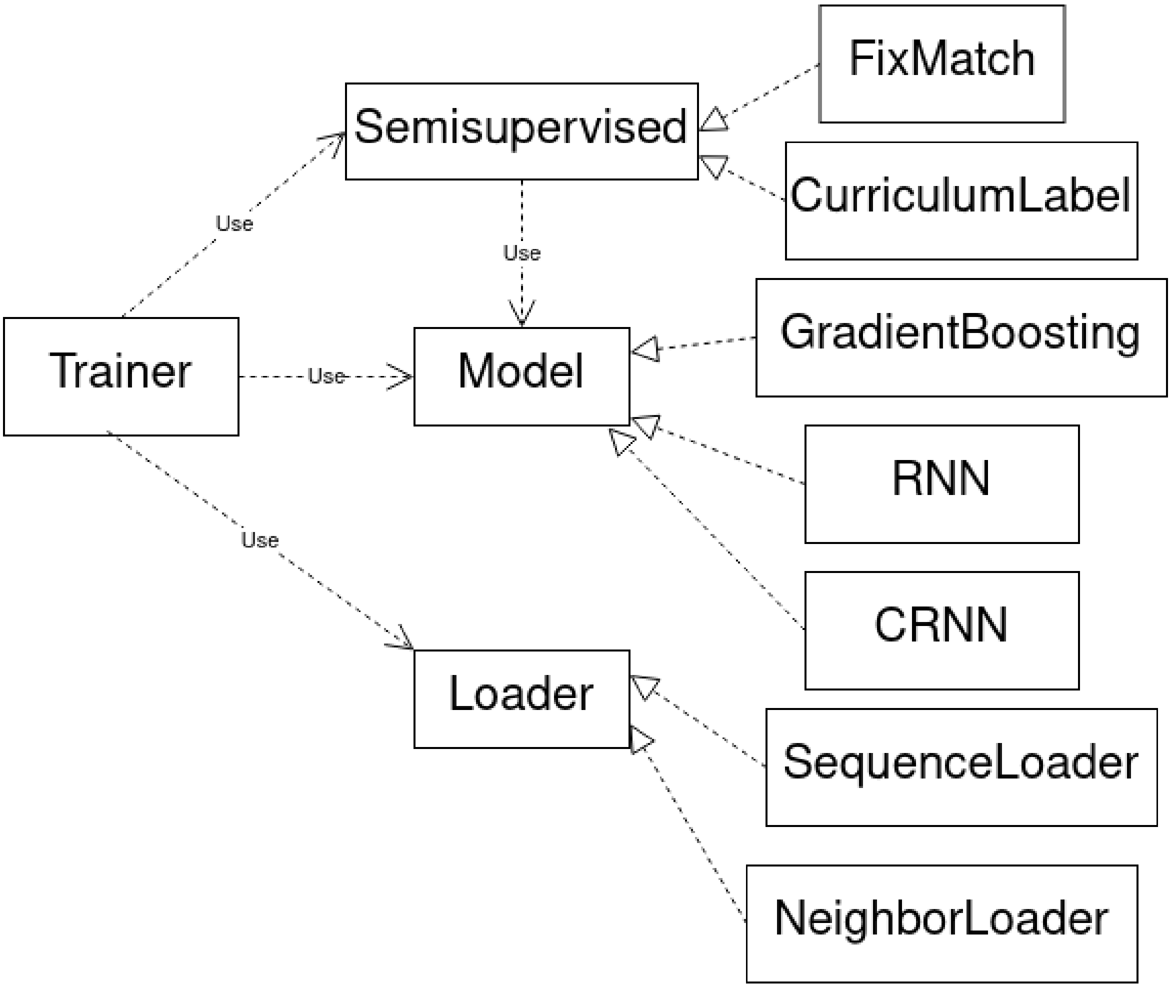


Figure 3.4. UML diagram of classes that are used for training the model.

is used to obtain frame probabilities. Finally, probabilities are converted to binary values based on the threshold value and the frames are merged into intervals and saved to a CSV file. Inference hash, model type, model version, and duration of the recording are saved to a CSV file for additional information.

To check the behavior of the model, spectrograms can be visualized with marked ground truth and predicted bowel sound intervals. The visualization module uses the recording, ground truth CSV file, and predicted CSV file. The example visualized spectrogram is shown in figure 3.5. The red-filling rectangles indicate the ground truth bowel sound intervals. The red border rectangles indicate the predicted bowel sound intervals. Augmentation transformations can be applied to the spectrogram before visualization for additional validation.

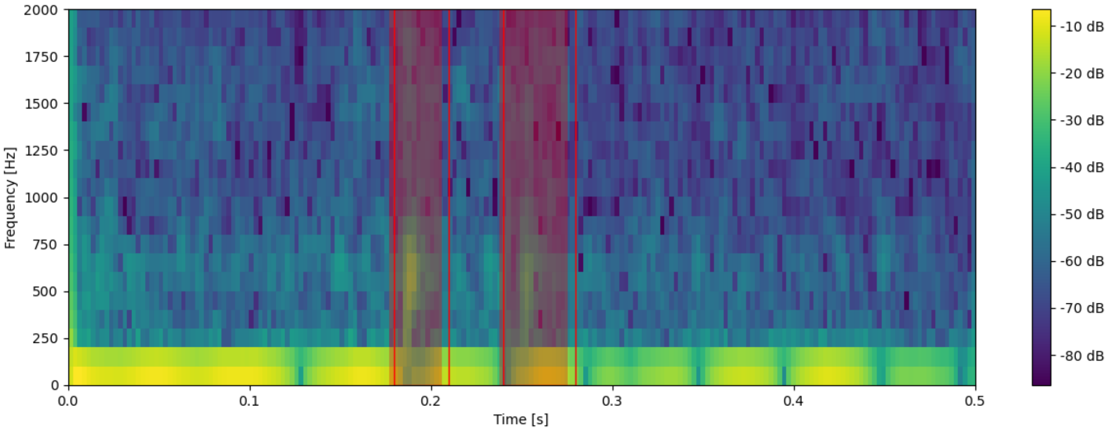


Figure 3.5. Example spectrogram with marked ground truth and predicted labels of bowel sounds.

3.7. Recordings analysis

Generated report of bowel sounds analysis contains statistics inspired by statistics used in the heart rate analysis [54]. The report is saved as an XLSX file containing a table of statistics and images of generated plots saved in separate tabs.

The following statistics and plots are calculated based on the CSV file of obtained bowel sound intervals:

- Total count of bowel sounds.
- Average of bowel sound occurrences per minute.
- Mean, standard deviation, median, minimum, maximum, 1st and 3rd quartile, 1st and 9th decile of bowel sound count per minute.
- Percentage of bowel sounds followed by another bowel sound within 50ms, 100ms, and 200ms.
- Mean, standard deviation, median, minimum, maximum, 1st and 3rd quartile, 1st and 9th decile of bowel sound duration.
- Root mean square of the successive differences (RMSSD).
- Natural logarithm of RMSSD.
- Standard deviation of bowel sound intervals (SDNN).
- Porta's index (PI).
- Guzik's index (GI).
- Plot of bowel sounds per minute.
- Plot of bowel sound average durations per minute.
- Histogram of bowel sounds per minute.
- Scatterplot of bowel sounds per minute and average durations per minute.

The examples of generated plots and report are shown in figures 3.6 and 3.7.

RRMSD calculates the average time between successive bowel sounds and is described by the formula (22), where RR$_i$ is the time duration between the ith and (i+1)th bowel

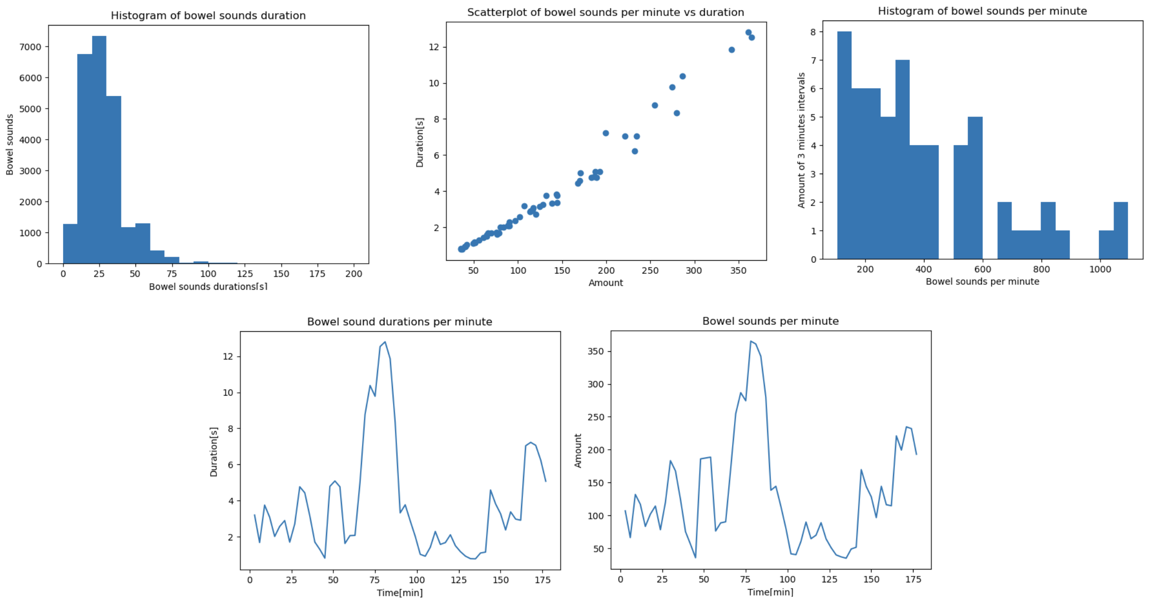
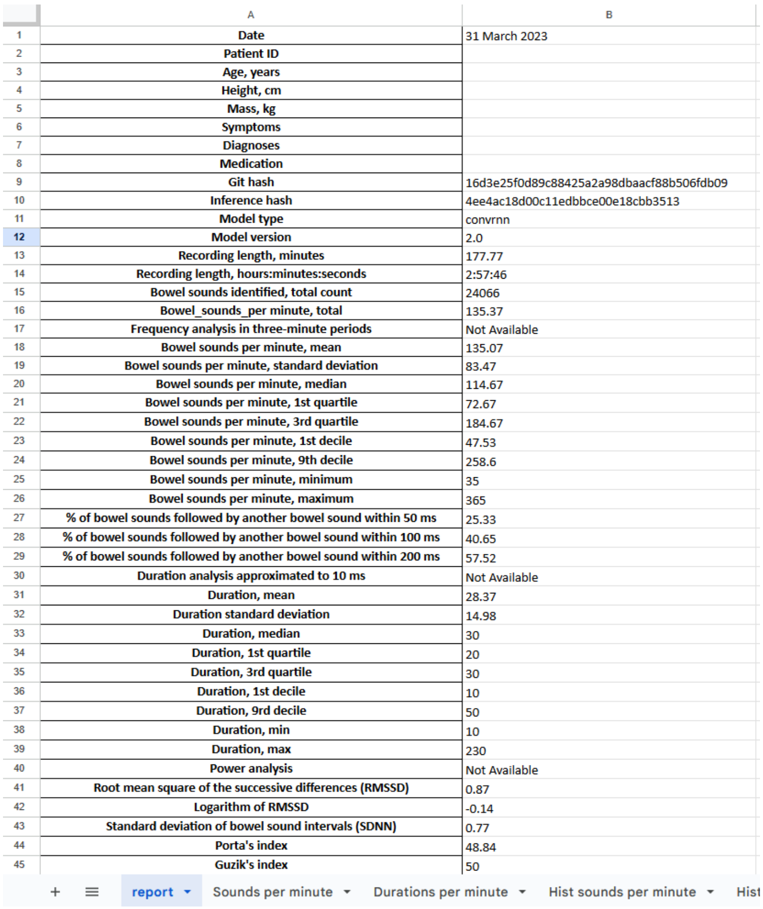


Figure 3.6. The plots generated from the obtained bowel sound time intervals from 3 hours long bowel sound recording.



	A	B
1	Date	31 March 2023
2	Patient ID	
3	Age, years	
4	Height, cm	
5	Mass, kg	
6	Symptoms	
7	Diagnoses	
8	Medication	
9	Git hash	16d3e25f0d89c88425a2a98dbaacf88b506fdb09
10	Inference hash	4ee4ac18d00c11edbbce00e18cbb3513
11	Model type	convrnn
12	Model version	2.0
13	Recording length, minutes	177.77
14	Recording length, hours:minutes:seconds	2:57:46
15	Bowel sounds identified, total count	24066
16	Bowel_sounds_per minute, total	135.37
17	Frequency analysis in three-minute periods	Not Available
18	Bowel sounds per minute, mean	135.07
19	Bowel sounds per minute, standard deviation	83.47
20	Bowel sounds per minute, median	114.67
21	Bowel sounds per minute, 1st quartile	72.67
22	Bowel sounds per minute, 3rd quartile	184.67
23	Bowel sounds per minute, 1st decile	47.53
24	Bowel sounds per minute, 9th decile	258.6
25	Bowel sounds per minute, minimum	35
26	Bowel sounds per minute, maximum	365
27	% of bowel sounds followed by another bowel sound within 50 ms	25.33
28	% of bowel sounds followed by another bowel sound within 100 ms	40.65
29	% of bowel sounds followed by another bowel sound within 200 ms	57.52
30	Duration analysis approximated to 10 ms	Not Available
31	Duration, mean	28.37
32	Duration standard deviation	14.98
33	Duration, median	30
34	Duration, 1st quartile	20
35	Duration, 3rd quartile	30
36	Duration, 1st decile	10
37	Duration, 9rd decile	50
38	Duration, min	10
39	Duration, max	230
40	Power analysis	Not Available
41	Root mean square of the successive differences (RMSSD)	0.87
42	Logarithm of RMSSD	-0.14
43	Standard deviation of bowel sound intervals (SDNN)	0.77
44	Porta's index	48.84
45	Guzik's index	50

Figure 3.7. Report generated from the obtained bowel sound time intervals from 3 hours long bowel sound recording.

sound.

$$RMSSD = \sqrt{\frac{\sum_{i=1}^{N} \mathrm{RR}_i^2}{N}} \tag{22}$$

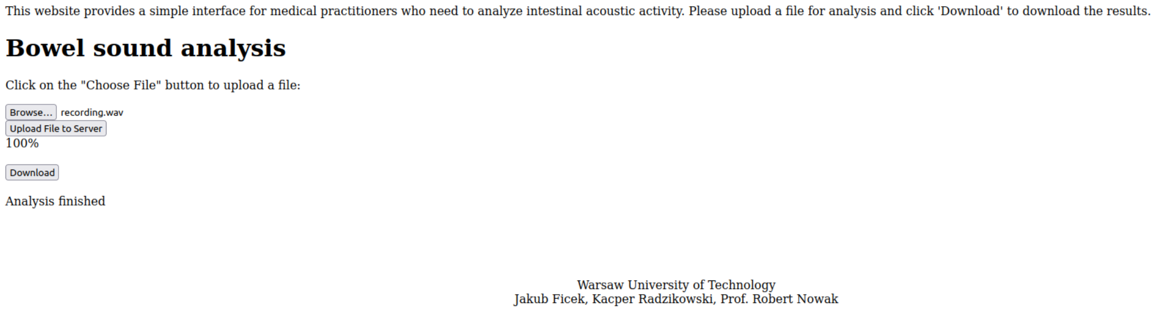
Porta's index and Guzik's index [55] measure the asymmetry of times between successive bowel sounds described by the formulas (23) and (24), where b is the number of bowel sounds for which time to the previous bowel sound is longer than time to the next bowel sound, l is the number of bowel sounds for which time to the previous bowel sound is shorter than time to the next bowel sound, m is the total number of bowel sounds, and D_i is the difference between time intervals of the (i-1)th and ith and the ith and (i+1)th bowel sound.

$$PI = \frac{b}{m-2} \times 100 \tag{23}$$

$$GI = \frac{\sum_{i=1}^{l} D_i}{\sum_{i=1}^{m-2} D_i} \times 100 \tag{24}$$

3.8. Website

To enable bowel sound analysis for doctors and researchers, I developed a website using the Flask framework in Python and JavaScript. It uses inference and reporting modules and allows users to upload the recordings, analyze them using the model, and download a compressed zip file containing a CSV file with bowel sound time intervals and an XLSX file with statistics and plots. The website runs in a Docker container using continuumio/miniconda3 as a base container. The website view is shown in figure 3.8.



This website provides a simple interface for medical practitioners who need to analyze intestinal acoustic activity. Please upload a file for analysis and click 'Download' to download the results.

Bowel sound analysis

Click on the "Choose File" button to upload a file:

100%

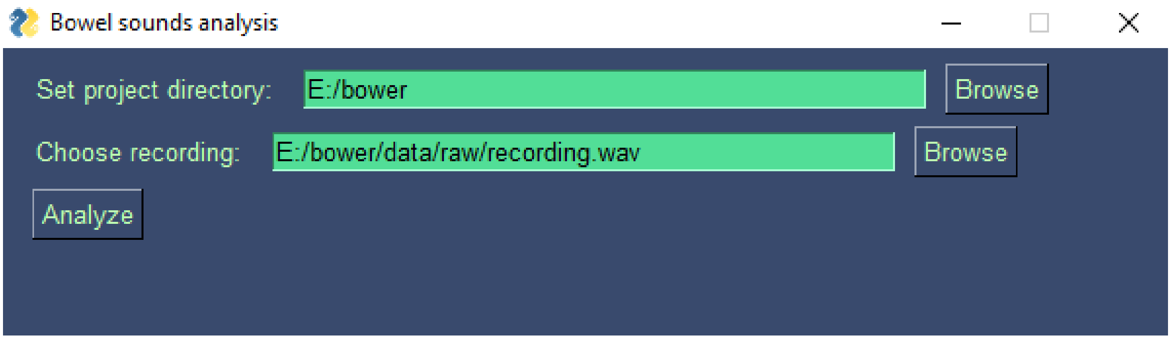
Analysis finished

Warsaw University of Technology
Jakub Ficek, Kacper Radzikowski, Prof. Robert Nowak

Figure 3.8. Website for bowel sound analysis.

3.9. Package

An additional packaged application with a simple graphical interface was developed to avoid the need for uploading big audio files to the website. The application was created using the PyInstaller library in Python, which automatically detects necessary dependencies and creates a single package. It enables running applications without installing Python and libraries. The graphical interface of the application for the Windows system is



Bowel sounds analysis

Set project directory: E:/bower

Choose recording: E:/bower/data/raw/recording.wav

Figure 3.9. GUI of the bowel sound analysis application on Windows system.

shown in figure 3.9. The user needs to choose the directory with the saved models and the recording to analyze. After analysis, the result files are saved in the project directory. The downside of this application is the big size of approximately 1.8GB. It is mainly associated with the necessity of packing the whole TensorFlow library into the application.

3.10. Conclusions

The software was written in Python. The neural network models were built using the TensorFlow library. Logistic regression, SVM, and random forest were built using the scikit-learn library and gradient boosting was built using the XGboost library. Librosa was used for audio processing. The website was developed using JavaScript and Flask. Other tools used in the application are NumPy, pandas, Matplotlib, PySimpleGUI, and Xlsxwriter.

The application consists of 31 files containing 2070 lines of code. I implemented 29 unit tests. The application was developed on Ubuntu 22.04 64bit operating system. The implementation of the application and inventing and testing of new methods took me approximately 500 hours.

4. New detection methods

4.1. Dataset and methodology

The dataset consists of 19 recordings of bowel sounds from different patients recorded during sleep. Recordings have been made using a modified Tascam DR-10CH recorder with a depth of 24 bits and frequency sampling of 44.1kHz. Each recording contains an associated CSV file with a list of time intervals of bowel sound occurrences annotated by an expert. Additionally, 6 long recordings without annotations are available. The total duration of annotated recordings is 54 minutes and the total duration of unannotated recordings is 11.6 hours.

In my engineering thesis, only two annotated recordings were available, so recordings were split into 2 seconds samples and mixed between datasets. This time whole recordings are split, so there are no samples from the same recording in the training and test dataset. The reason for this is to avoid the model learning features of particular recordings, instead of general bowel sound features and falsely assume better quality of the model during testing. The ratio between train, validation, and test dataset is approximately 70%, 15%, and 15% respectively. Additionally, recordings without annotations are used in the training process for semi-supervised learning methods. Cross-validation isn't used because there are not enough recordings to reliably split them between parts.

The best models are chosen using the accuracy of the validation dataset. Additionally, PR AUC and IOU based metrics are measured for additional information due to unbalanced data. For neutral networks, the batch size is set to 8 and the model finished training using an early stopping strategy based on validation loss after 20 epochs. Adam algorithm implemented in Keras library is selected for parameters optimization. The presented results come from the best epoch based on the validation accuracy.

Table 4.1 shows bowel sound statistics of different recordings and the division between datasets. The durations of recordings are very different. The longest is 14 minutes 25 seconds and the shortest is less than 6 seconds. a0.wav has a very different average length and standard deviation from other recordings because it contains annotations of different types of bowel sounds than single burst bowel sounds. To avoid this inconsistency, bowel sounds other than single burst bowel sounds are discarded from annotations. a.wav shows updated parameters for only single burst bowel sounds. We can see, that ratio and count of bowel sounds are very different in different recordings. Partly, it is because the frequency of bowel sounds varies depending on the subject and period of digestion during recording. Another reason is that there is some inconsistency in annotating different recordings. For example, only more distinct bowel sounds were annotated in b.wav and d.wav recordings. For this reason, the train dataset contains a lower ratio of bowel sounds, so models trained on the train dataset may achieve lower recall on the validation dataset. The average length

Table 4.1. Parameters of bowel sounds in the individual recordings. Count is the count of bowel sounds in the recording, Ratio is the ratio of the summed duration of bowel sounds to the total duration of the recording. Average and SD are the average and standard deviation of duration of individual bowel sounds.

Recording	Duration [s]	Count	Ratio [%]	Average [ms]	SD [ms]
a0.wav	855.22	1984	15.62	67.35	167.24
Train dataset					
a.wav	855.22	1758	6.31	30.70	9.22
b.wav	701.76	402	1.34	23.46	24.15
c.wav	360.06	401	2.44	21.88	7.05
d.wav	302.85	236	1.52	19.50	4.65
e.wav	63.18	280	9.25	20.86	5.01
All	**2283.08**	**3077**	**3.62**	**25.33**	**12.75**
Validation dataset					
f.wav	238.97	1067	13.81	30.93	10.90
g.wav	103.59	116	1.95	17.43	27.01
h.wav	46.18	52	2.20	19.56	4.50
i.wav	30.89	75	8.35	34.41	32.87
j.wav	23.37	51	4.93	22.61	4.51
k.wav	19.12	38	3.67	18.47	3.89
l.wav	5.95	56	19.87	21.11	5.56
All	**468.07**	**1455**	**8.90**	**26.00**	**14.61**
Test dataset					
m.wav	240.00	1035	13.44	31.17	10.91
n.wav	158.83	222	2.61	18.74	5.18
o.wav	21.79	40	3.78	20.60	8.18
p.wav	19.28	107	11.64	20.97	6.28
q.wav	16.84	44	3.89	14.88	2.22
r.wav	16.01	57	6.36	17.86	5.23
s.wav	6.56	113	30.54	17.74	4.12
All	**479.31**	**1618**	**9.01**	**24.96**	**8.11**

and standard deviation of bowel sounds are similar across different recordings and are consistent with characteristics of single burst bowel sounds described in section 2.1.

4.2. Results of the base model on new data

The purpose of this experiment is to compare the best results from my previous work to the results of models trained using the new data.

Previously developed CRNN model contains 486281 parameters and its detailed architecture is shown in table 4.2.

Table 4.3 shows the results of the developed models. CRNN1 shows the results from my engineering thesis [1] by training the model on 2 recordings. CRNN2 shows the results of the model trained on 19 recordings using the previous division method with mixing

Table 4.2. CRNN model layers overview.

Layer type	Parameters	Output shape
Input	0	(200,20,4)
Conv2D(kernel_size=(3,3), activation=ReLU)	300	(200,18,2,30)
Conv2D(kernel_size=(4,2), activation=ReLU)	14460	(200,15,1,60)
Dropout(rate=0.4)	0	(200,900)
BidirectionalGRU(activation=ReLU)	471360	(200,160)
Dropout(rate=0.4)	0	(200,160)
Dense(activation=Sigmoid)	161	(200,1)

samples between training and test dataset. CRNNBase shows results with the new dataset division described in the previous section. CRNNBase training data is smaller because the CRNN2 model didn't use a validation dataset (there was no need because the model was already chosen). We can see that after applying the new division, the F1 score decreases by a few percentage points. It is related to less sensitive annotation of some train recordings which can be seen by lower recall. Additionally, some part of the decrease comes from the previous models learning features of particular recordings which is impossible using the new division.

Table 4.3. Results of the CRNN model using different dataset sizes and divisions.

Model	Train data [s]	Ratio [%]	Acc [%]	Precision [%]	Recall [%]	F1 [%]
CRNN1	920	84.80	93.41	57.57	85.63	75.98
CRNN2	2570	94.15	97.73	82.70	77.30	79.92
CRNNBase	2278	91.33	95.81	82.24	65.95	73.20

4.3. Sliding window method

The new approach for classifying frames tested in this work is to classify each frame individually without passing the whole sequence. To allow the model to get more information, neighboring data is added to the model depending on the context window size. The method of dividing frames into windows and context windows is shown in figure 4.1.

The advantage of this method is the lower computation complexity of individual frame prediction because there is no need to propagate through a sequence of frames. The downside is that there is big redundancy related to taking the same parts of recordings into context windows of adjacent frames when using a bigger context size. Additionally, input dimension and model complexity grow fast with increasing context size, so taking a smaller context window size may result in the model not having enough resolution to make the correct prediction.

The purpose of this experiment is to determine if the sliding window approach can outperform the sequence approach. 4 different machine learning models are tested

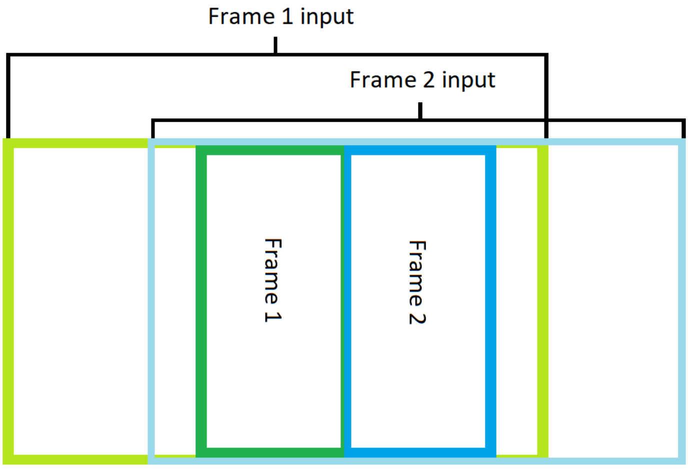


Figure 4.1. Sliding window method. The dark green square represents the first frame and the light green represents the context of the first frame that enters as input to the model. The dark blue square represents the second frame and the light blue square represents the context of the second frame.

with hyperparameters adjustment: logistic regression, SVM, random forest, and gradient boosting.

4.3.1. Logistic regression

Table 4.4 shows metrics achieved using logistic regression depending on the context size. The best developed model uses a 250ms context size. We can see that AUC PR on the train and validation datasets is poor and the models are underfitting. Hence, we can deduce that nonlinear operations are needed for good classification.

Table 4.4. Results of the logistic regression models on train and validation datasets depending on the context size

Context size [ms]	Train Acc [%]	Train AUC PR [%]	Acc [%]	AUC PR [%]
10	93.15	10.73	91.63	13.82
20	93.68	17.82	92.04	17.94
50	94.42	27.56	92.36	20.18
100	94.69	31.18	92.38	20.18
150	94.75	32.11	92.46	20.96
200	94.86	33.43	92.50	21.46
250	**94.93**	**34.28**	**92.52**	**21.68**
300	94.98	34.97	92.43	21.01
350	95.01	35.44	92.40	20.83
400	95.03	35.76	92.40	20.90
450	95.07	36.29	92.33	20.52
500	95.12	36.92	92.35	20.98

4.3.2. Support vector machine

SVM with radial basis kernel was tested. Table 4.5 shows achieved metrics depending on the context size. The best model uses a 50ms context size. A big problem with using SVM with radial basis kernel is that the computational cost of training grows fast with the number of samples and features, so a big context size isn't tested. Results are better than logistic regression, but the models are still underfitting.

Table 4.5. Results of SVM models on train and validation datasets depending on the context size.

Context size [ms]	Train Acc [%]	Train AUC PR [%]	Acc [%]	AUC PR [%]
10	93.92	20.09	92.06	16.46
50	**96.12**	**48.21**	**92.83**	**24.48**
100	96.19	49.15	92.52	21.19
150	95.96	46.86	92.35	19.41

4.3.3. Random forest

Table 4.6 shows metrics of developed random forest models depending on context size and the maximum depth of tree estimators. For all experiments, number of estimators is set to 100. The results are much better than logistic regression and SVM. The best model uses 150ms context size and 70 maximum depth. We can see that the random forest model tends to overfit, especially for bigger max depth.

Table 4.6. Results of random forest models on train and validation datasets depending on the context size and maximum depth using 100 estimators.

Context size [ms]	Max depth	Train Acc [%]	Train AUC PR [%]	Acc [%]	AUC PR [%]
10	10	94.36	25.73	92.53	21.25
50	20	97.25	63.80	93.32	29.63
50	30	98.18	76.02	93.54	31.93
50	35	98.57	81.14	93.53	31.96
100	35	98.43	79.31	93.48	31.31
100	40	98.75	83.59	93.55	32.10
100	45	99.05	87.56	93.48	31.38
150	60	99.52	93.71	93.56	32.19
150	**70**	**99.67**	**95.65**	**93.58**	**32.32**
150	80	99.77	97.03	93.52	31.72
200	70	99.62	95.00	93.47	31.21
200	80	99.73	96.51	93.49	31.46
200	90	99.82	97.67	93.45	31.01
250	90	99.80	97.39	93.48	31.26
250	100	99.86	98.18	93.45	30.93
300	90	99.78	97.05	93.36	30.07
300	100	99.77	97.05	93.31	30.07

4.3.4. Gradient boosting

Table 4.7. Results of gradient boosting models on train and validation datasets depending on context size using 100 estimators and maximum depth of 6.

Context size [ms]	Train Acc [%]	Train AUC PR [%]	Acc [%]	AUC PR [%]
10	95.90	46.13	92.92	26.60
50	97.84	71.49	93.84	35.13
100	98.06	74.33	93.77	34.48
150	98.26	76.99	93.77	34.55
200	98.24	76.73	93.85	35.26
250	98.32	77.77	93.80	34.92
300	**98.34**	**78.04**	**93.86**	**35.43**
350	98.45	79.49	93.81	34.91

Table 4.8. Results of gradient boosting models on train and validation datasets depending on maximum depth using 100 estimators and 300ms context size.

max depth	Train Acc [%]	Train AUC PR [%]	Acc [%]	AUC PR [%]
4	96.86	58.77	93.49	31.59
6	**98.34**	**78.04**	**93.86**	**35.43**
8	99.86	98.14	93.78	34.51
10	100.00	100.00	93.69	33.63

Table 4.9. Results of gradient boosting models on train and validation datasets depending on an number of estimators using 300ms context size and maximum depth of 6.

n_estimators	Train Acc [%]	Train AUC PR [%]	Acc [%]	AUC PR [%]
50	97.41	65.84	93.80	34.80
100	**98.34**	**78.04**	**93.86**	**35.43**
150	99.11	88.24	93.83	35.13

Gradient boosting with tree estimators was tested. Tables 4.7, 4.8, 4.9 show metrics of developed gradient boosting models depending on the context size, estimators number, and maximum depth. The best developed model uses a 300ms context size, 100 estimators, and 6 maximum depth.

Gradient boosting handles overfitting better and achieves higher accuracy and PR AUC than random forest, but it is still much worse than the previously developed CRNN model. Probably, it is because the relation between frames is important, so the sliding window method achieves much lower metrics than the methods based on the sequence of frames. It could be partly resolved by using a very big context size (comparable to the duration of a sequence of frames equal to 2 seconds), but it makes a big redundancy and increases computational cost a lot. Results could be also improved by using a convolutional neural network or regularization methods like L2 loss or data augmentation, but it is unlikely to outperform the sequence of frames approach, so the sliding window approach isn't tested further. IOU F1 scores for models using the sliding window approach are shown in figure 4.2.

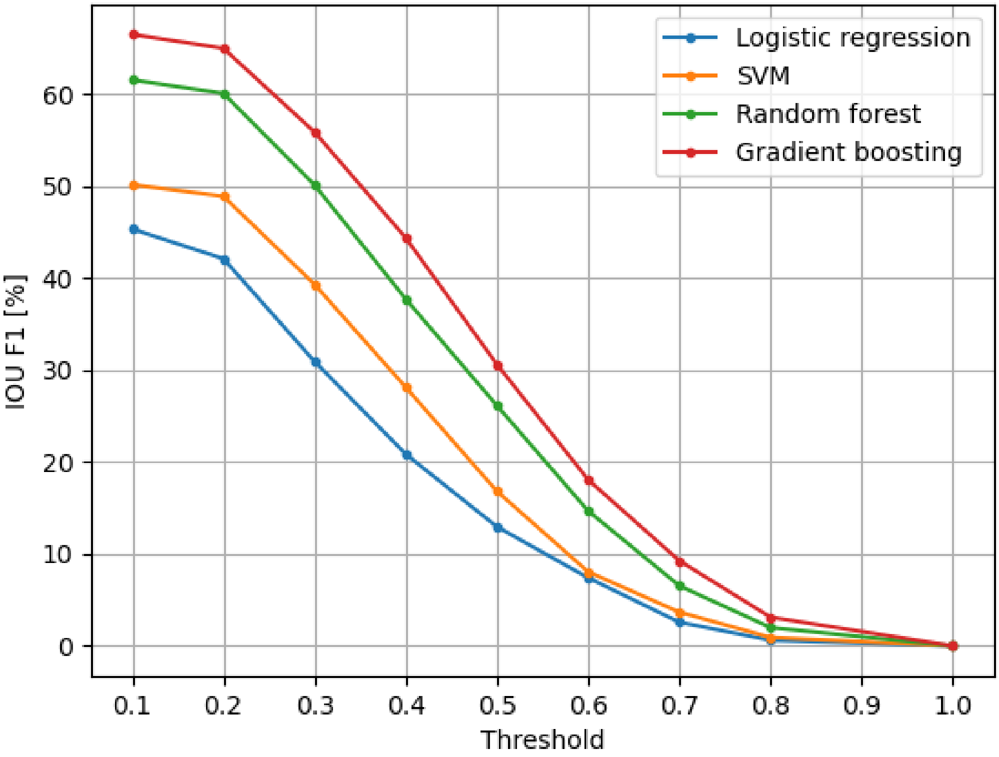


Figure 4.2. IOU F1 score models developed using sliding window approach for a given threshold.

4.4. Focal loss

Another method of improving the model is to use the focal loss for the model training, which is reported to achieve better scores for the unbalanced datasets [56].

$$FL(p) = -\alpha(1-p)^{\gamma} log(p) \tag{25}$$

The formula is shown in (25). Focal loss is a modification of binary cross entropy that focuses more on miss-classified examples. It is achieved by the γ parameter, which reduces the cost of an error on well classified examples when the probability of ground truth class obtained from the model is high which is shown in figure 4.3. α parameter is used to account for the imbalance of the dataset. I decided to test this loss type because there are much fewer frames with bowel sound occurrence compared to silent frames in the dataset.

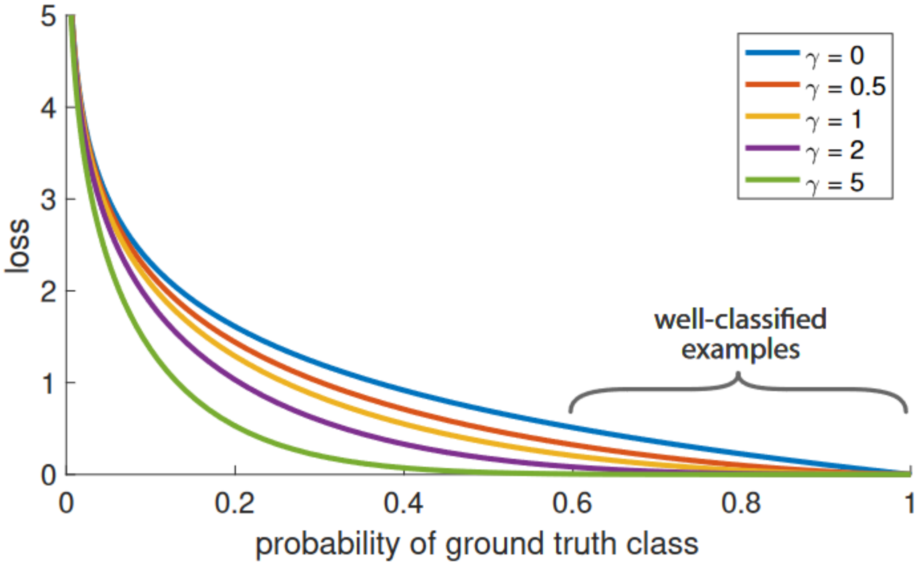


Figure 4.3. Focal loss and binary cross-entropy functions comparison adapted from [56].

The purpose of this experiment is to determine if using focal loss instead of binary cross entropy can benefit the model. Focal loss parameters in this experiment are set to $\alpha = 0.25$, $\gamma = 2$.

Table 4.10. Loss function comparison using CRNN model on train and validation dataset.

Loss function	Train Acc [%]	Train AUC PR [%]	Acc [%]	AUC PR [%]
Cross entropy	**98.42**	**84.93**	**95.81**	**78.91**
Focal Loss	99.45	99.41	95.22	79.06

Table 4.10 summarizes the impact of using focal loss for the CRNN model. We can see that metrics on the train data set are much higher using focal loss, but they decrease significantly on the validation dataset. A possible explanation for this is the different sensitivity of annotations in training and validation datasets and the tendency of focal loss to punish every miss-classfied example more severely. This leads to model learning the annotation inconsistency and makes the model overfit, so the model quality is worse on the validation dataset. For this reason, a focal loss isn't used but with less noisy annotations, the focal loss could lead to improving the model and it is potentially a good idea to use it in bowel sound detection task.

4.5. Frame size comparison

The downside of the method based on the frame division, compared to the filter methods, is that the precision of the acquired time intervals is limited to the frame size. Certain sizes of frames are limiting the possible IOU based metrics. It is related to two types of errors. The first one is related to merging two bowel sounds into one when the time distance between two sounds is smaller than the frame size. One way to solve this problem would be annotating the first frame of bowel sound differently (Inside–outside–beginning tagging), similar to the named entity recognition task in natural language processing. It isn't tested in this work, because the error is rare and the achieved scores aren't very close to the theoretical best, so there is still a place to improve the model with standard tagging. The second error is related to approximating time intervals to the whole frames which lower the IOU scores with a high enough threshold. Both of these errors can be reduced by making the frame size smaller, which is presented in figure 4.4. The figure shows the IOU F1 score for different thresholds with a given frame size, assuming that the model predicts every frame correctly. We can see, that the theoretical best score improvement using smaller frame sizes is significant for higher thresholds.

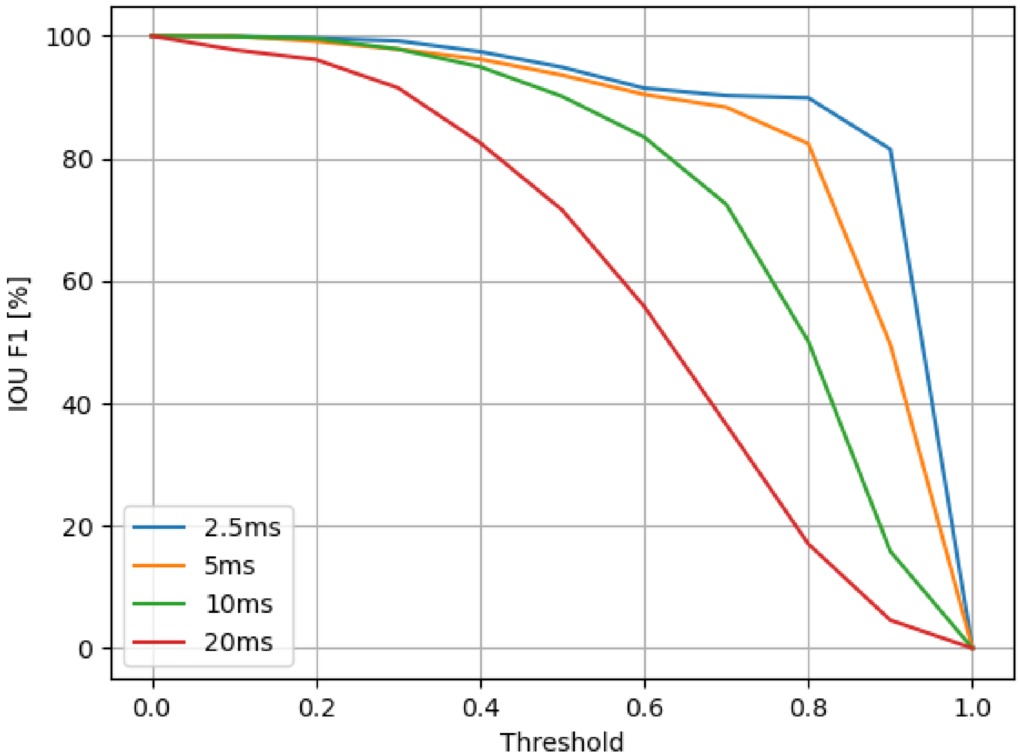


Figure 4.4. Theoretical best IOU F1 score for given threshold based on the used frame size.

Additionally, decreasing frame size allows us to get more accurate statistics of recording from obtained bowel sounds time intervals. For these reasons, testing frame sizes smaller than 10ms is valuable.

The purpose of this experiment is to test the impact of different sizes of frames on the quality of the model. It makes little sense to use convolutional layers to get the features vector of the frame because using a very small frame size leads to a low dimension size of a single frame. For this reason, convolutional layers were replaced with dense layers. Additionally, after initial testing, a second GRU layer was added to the model. The overall architecture of the RNN model is presented in table 4.11 and it contains 449201 parameters.

Table 4.11. RNN model layers overview.

Layer type	Parameters	Output shape
Input	0	(400,40)
Dense(activation=ReLU)	4920	(400,120)
Dense(activation=ReLU)	29040	(400,240)
Dense(activation=ReLU)	86760	(400,360)
Dropout(rate=0.4)	0	(400,360)
BidirectionalGRU(activation=ReLU)	212160	(400,160)
BidirectionalGRU(activation=ReLU)	116160	(400,160)
Dropout(rate=0.4)	0	(400,160)
Dense(activation=Sigmoid)	161	(400,1)

Table 4.12 shows metrics of the developed RNN model depending on the frame size and context size (equal to the sequence length multiplied by frame size). We can see that the best metrics are achieved by a 5ms frame size and a 2000ms context size. 2.5ms frame size also achieved comparably good results. 10msthe frame size achieved much lower scores probably because frame dimension size is higher, so it is better to use convolutional layers and the CRNN model. The bigger context size isn't tested, because it leads to increasing the length of a sequence of frames to more than 400. RNN models aren't good with very long sequences and it also significantly increases training time.

Table 4.12. Results of the RNN model on the validation dataset depending on the window frame size and context size.

Window size [ms]	Context size [ms]	Acc [%]	AUC PR [%]
10	2000	95.53	78.82
5	1000	95.78	81.09
5	**2000**	**95.89**	**82.00**
2.5	500	95.86	82.62
2.5	1000	95.87	82.54

4.6. Data augmentation

Data augmentation methods for audio data need to be different than for images. For example, rotation or translation methods don't make sense for spectrograms. Augmentation methods of spectrograms for audio analysis were tested in [57]. Based on this work and bowel task characteristics, I decided to implement the following data augmentation methods:

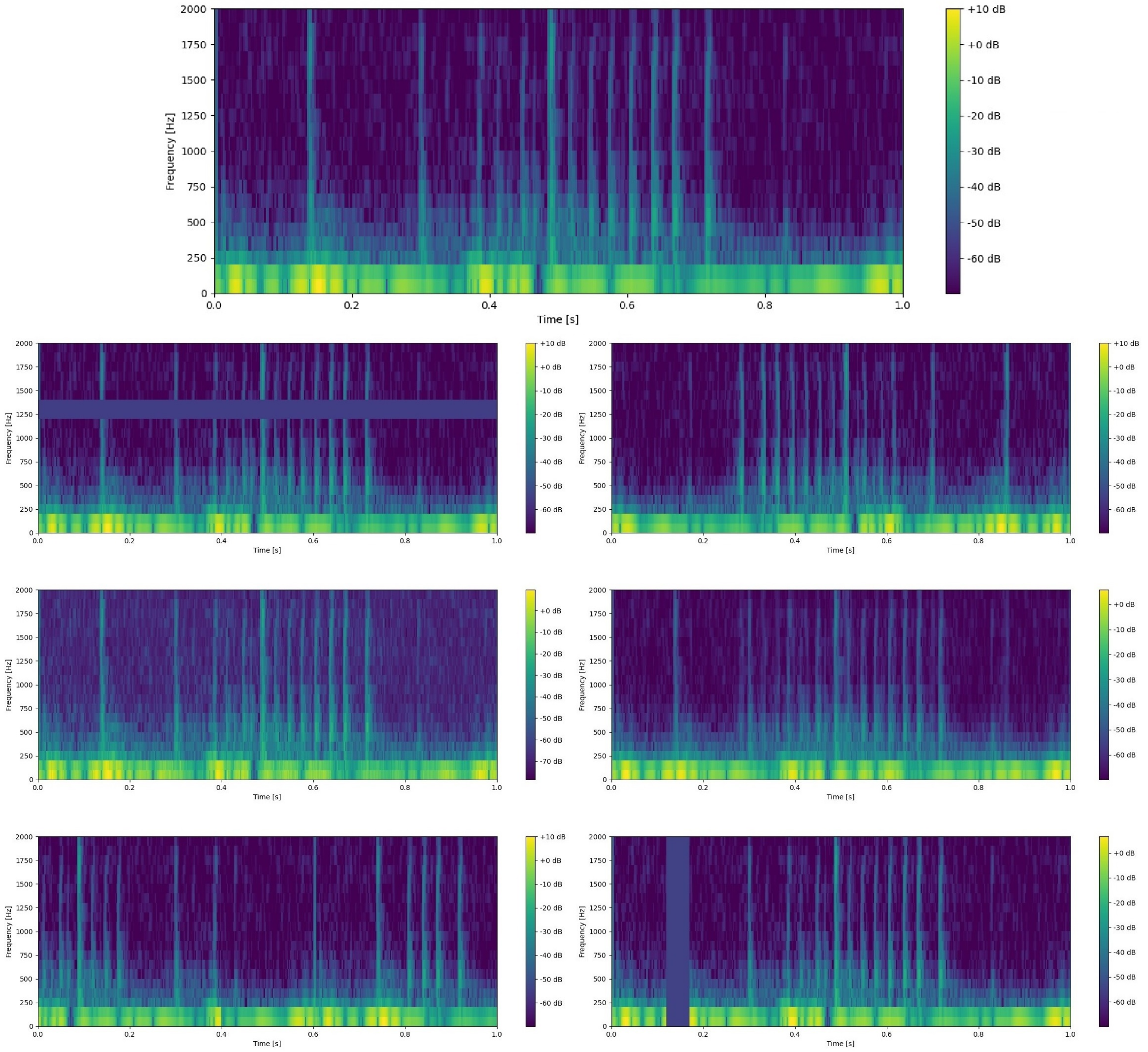


Figure 4.5. Different augmentation methods comparison. The top image shows the original spectrogram. The top left image shows frequency masking augmentation. The top right image shows vertical flip augmentation. The middle left image shows Gaussian noise augmentation. The middle right image shows MixUp augmentation with flipped spectrogram using $\beta = 0.77$. The bottom left image shows shuffle augmentation using 5 intervals. The bottom right image shows time masking augmentation.

- Overlapping (Overlap) - involves taking successive spectrograms from the recording using smaller steps with overlapping.
- Vertical flip (Flip) - involves reversing the time axis of the spectrogram. It is reasonable because single burst bowel sounds spectrograms are quite symmetrical.
- Frequency mask (Freq) - involves setting the values of the spectrogram to 0 for a random frequency range (values are set to 0 instead of the mean because spectrograms are standardized using statistics obtained from the training data).
- Time mask (Time) - involves setting the values of the spectrogram to 0 for a random time range.

- Gaussian noise (Gauss) - involves adding Gaussian noise with a certain variance to the values of the spectrogram.
- MixUp - involves taking two different sequences of frames and weight averaging spectrograms and labels. Described in detail in section 2.5.
- Shuffle - involves dividing a sequence of frames into intervals and shuffling them randomly.
- Permutation - involves shuffling the order of bowel sounds leaving the silent parts unmodified. It was found useful for the classification of environmental sounds [58].

The mentioned data augmentation methods are visualized using an example spectrogram in figure 4.5.

The purpose of this experiment is to test the impact of different augmentation methods on the quality of the model. Firstly, a single augmentation methods are tested to determine which augmentation methods are useful in the task of the detection of bowel sounds. Then, combinations of various chosen augmentation methods are tested. All experiments are executed using the CRNN model. All augmentation methods are applied to time domain signals or spectrograms, so the conclusions should be kept when using other similar neural network architectures. The augment ratio describes how much augmented data there is compared to standard data (0.00 means there was no data augmentation).

For testing a single augmentation methods, augmented data is created by applying the augmentation method one or more times on the whole train dataset.

Table 4.13 shows the metrics using different augmentation methods. The biggest improvement is achieved using overlapping, flip, MixUp, and shuffle methods. Frequency mask, time mask, Gaussian noise, and permutation improve the model slightly or make the model worse, so they aren't tested in experiments using combinations of augmentation methods.

For the combination of augmentation methods, the size of the whole augmented data is defined. Then, to create an augmented sample, the sample is taken from the standard train dataset and each augmentation method is applied with a given probability. To generate a MixUp sample, two samples with applied augmentation methods are randomly taken.

Table 4.14 shows metrics for applying combinations of augmentation methods, where Type describes the augmentation methods followed by the probability of using a given augmentation method (Flip0.5Shuffle0.8 means that approximately 50% of the augmented data is flipped and 80% of augmented data is shuffled). The best parameters for augmentation methods from previous experiment are used. The best results are obtained using a combination of overlapping and MixUp augmentation. We can also see that the use of many augmentation methods doesn't lead to further improvement of the model. When the data is modified too much, the model isn't able to learn to predict the augmented samples.

Table 4.13. CRNN model results on the validation dataset depending on the applied augmentation method.

Type	Parameter	Augment Ratio	ACC [%]	AUC PR [%]
Base	**None**	**0.00**	**95.81**	**78.91**
Overlap	overlap=50%	1.00	95.83	79.76
Overlap	overlap=66.6%	2.00	95.86	80.12
Overlap	**overlap=75%**	**3.00**	**95.93**	**79.81**
Overlap	overlap=80%	4.00	95.83	79.16
Flip	**None**	**1.00**	**95.93**	**81.25**
Freq	size=5%	1.00	95.84	78.52
Freq	size=5%	2.00	95.77	78.65
Freq	size=10%	1.00	95.67	78.76
Time	size=5%	1.00	95.81	76.73
Time	size=10%	1.00	95.71	77.91
Gauss	**SD=5%**	**1.00**	**95.86**	**77.49**
Gauss	SD=10%	1.00	95.74	77.91
Gauss	SD=30%	1.00	95.63	77.47
MixUp	$\alpha = 0.2$	1.00	95.85	79.30
MixUp	$\alpha = 0.2$	2.00	95.41	75.40
MixUp	**$\boldsymbol{\alpha} = 8$**	**1.00**	**95.94**	**78.91**
MixUp	$\alpha = 8$	2.00	95.84	80.35
MixUp	$\alpha = 32$	1.00	95.92	78.61
Shuffle	intervals=5	1.00	95.82	79.26
Shuffle	intervals=5	2.00	95.85	79.90
Shuffle	intervals=5	3.00	95.78	79.04
Shuffle	intervals=10	1.00	95.81	79.25
Shuffle	intervals=10	2.00	95.84	79.94
Shuffle	**intervals=10**	**3.00**	**95.86**	**80.27**
Permutation	None	1.00	95.79	79.42
Permutation	None	2.00	95.76	80.52

Table 4.14. CRNN models results on the validation dataset depending on the applied combination of augmentation methods.

Type	Augment Ratio	ACC [%]	AUC PR [%]
Base	0.00	95.81	78.91
Flip0.5Mixup0.5	2.00	95.93	78.87
Flip0.8Shuffle0.8Mixup0.5	4.00	95.82	78.46
Overlap0.5Mixup0.5	**2.00**	**95.95**	**79.28**
Flip0.33Overlap0.33MixUp0.33	3.00	95.90	80.30
Gauss0.3Overlap0.5Shuffle0.2Flip0.8	5.00	95.79	80.31
Gauss0.4MixUp0.4Shuffle0.5Flip0.8	5.00	95.72	79.69

4.7. Curriculum labeling

Curriculum labeling is a semi-supervised learning method that uses a pseudo-labeling strategy introduced in [59]. The method overview is shown in figure 4.6. It involves training multiple models from scratch in sequence. The most confident predictions on unlabeled samples using the current model are added to the training data for training the next model. The models are trained until the whole unlabeled data will be used for training the final model. In my task, the sequence approach is used, so the lowest certainty from individual frames is used to determine the sequence prediction certainty. The main disadvantage of this method is the big computation cost of training many models.

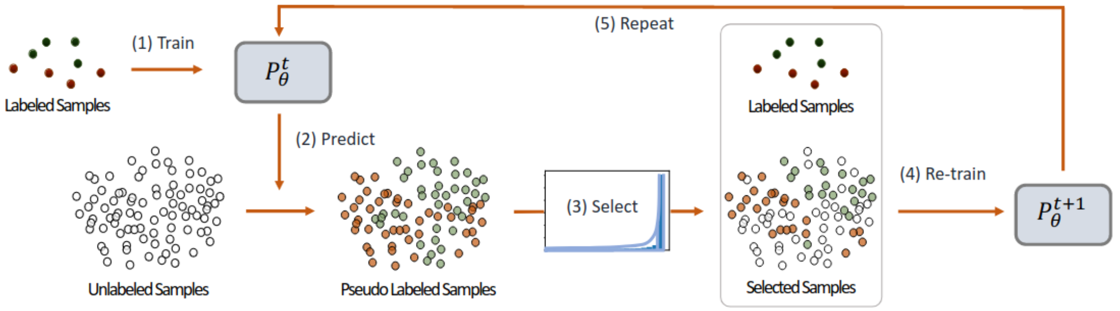


Figure 4.6. Scheme of curriculum labeling method from [59].

The purpose of this experiment is to test curriculum labeling semi-supervised method. The experiment uses the base CRNN model to predict the labels on unlabeled data. Sequences with no frames with probability output in the range [0.4,0.6] are added to the training data of the next iteration training. The certainty threshold is set low because the most certain sequences are the parts with no bowel sounds. Setting the certainty threshold higher results in adding only sequences with no bowel sounds and making the train dataset more imbalanced which results in worse results. Table 4.15 shows the final results for applying the curriculum labeling method, where the Pseudo-label ratio is the ratio of pseudo-labeled data to the standard data during training. We can see that curriculum labeling obtained good results and achieved 0.12% accuracy improvement

Table 4.15. CRNN models results on the validation dataset using curriculum labeling method.

Train iteration	Pseudo-label Ratio	Training ratio	ACC [%]	AUC PR [%]
0	0.00	96.44	95.81	78.91
1	2.74	98.18	95.88	80.16
2	6.24	97.53	95.90	80.22
3	9.76	96.77	95.90	80.00
4	11.78	96.31	95.92	80.39
5	**12.76**	**96.04**	**95.93**	**80.15**

over the base model after 5 iterations. In the first iterations, the train dataset becomes more imbalanced mainly because sequences with fewer bowel sounds are classified as the most certain but after more iterations, the ratio becomes close to the standard train data ratio.

4.8. Fixmatch

Fixmatch is a commonly used semi-supervised learning method that uses a combination of pseudo-labeling and consistency regularization strategies introduced in [60]. The method overview is shown in figure 4.7. It involves training the model on training data and then obtaining predictions on the unlabeled data transformed using weak augmentation. Then, the most confident subset of unlabeled data is transformed using strong augmentation and added to the training data. The new model is trained using an augmented dataset. In [60], flip and shift methods are used as weak augmentation and RandAugment [61] is used as strong augmentation which involves methods like masking, shearing, and changing brightness.

In my task, similar to the curriculum labeling approach, the lowest certainty from the sequence is used to obtain the sequence certainty. Flipping and Gaussian noise is tested as weak augmentation and Gaussian noise, frequency masking, and time masking is tested as strong augmentation. Overlapping can't be used because it doesn't directly modify the samples. Shuffle, permutation, and MixUp aren't trivial to implement for FixMatch, because these augmentation methods affect the labels of the sequence non-deterministically, so the implementation would require saving and applying the mapping of labels between weak and strong augmentation for each sample. For this reason, these augmentation types aren't tested in this thesis, but it is a potential path for future work.

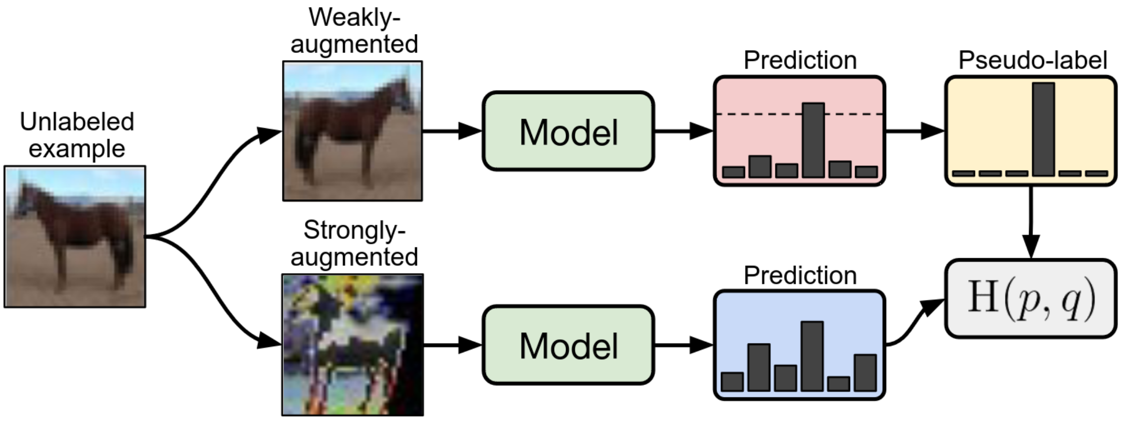


Figure 4.7. Scheme of Fixmatch method from [60].

The purpose of this experiment is to test FixMatch semi-supervised learning method with weak and strong augmentation methods mentioned in the previous paragraph. Sequences with no frames with probability output in the range [0.45,0.55] are added to the training data. The threshold certainty is low to avoid the problem of a more imbalanced dataset. Additionally, labeled data is augmented using the best augmentation methods combination (Overlap and MixUp).

Table 4.16. CRNN models results on the validation dataset using FixMatch method depending on the applied augmentation methods.

Weak	Strong	Pseudo-label Ratio	ACC [%]	AUC PR [%]
None	None	0.00	95.81	78.91
None	None	6.13	95.90	79.87
Gauss	Freq,Time	5.74	95.91	80.10
Flip	**Freq,Time**	**9.11**	**96.09**	**81.80**
Flip	Freq,Time,Gauss	9.11	96.02	80.18

Table 4.16 shows the results obtained using the FixMatch method. In the first two experiments, no augmentation method is used to obtain the base results. The best results are obtained using vertical flip as weak augmentation and frequency and time masking as strong augmentation. Using flip instead of Gaussian noise as weak augmentation seems crucial because it significantly improves the model. Probably Gaussian noise with a 5% standard deviation affects the samples too much to be used as a weak augmentation.

4.9. New detection models summary

Tables 4.17 and 4.18 show a summary of the results of all developed models. RNN model with a 5ms frame size and 2000ms context size is used. The combination of overlapping and MixUp augmentation is used for augmented models. Fixmatch with vertical flip as weak augmentation and frequency and time masking as strong augmentation is used for semi-supervised models.

The best model is CRNN using semi-supervised learning. The RNN model improves less from semi-supervised learning and data augmentation partly because the model has a smaller capability to fit the increased training data than the CRNN model. Potentially, increasing the size of RNN could improve the metrics. Additionally, all tests to adjust parameters of data augmentation and semi-supervised learning were done using the CRNN model, so they may be suboptimal for the RNN model.

4.10. Test dataset results

The semi-supervised versions of CRNN (CRNNFinal) and RNN (RNNFinal) models were trained on merged training and validation datasets and tested using a test dataset. Table 4.19 shows the acquired results compared to the CRNN1 result from the engineering thesis. The final models improved which is related to using additional data from the validation set during training. The final CRNN model achieves higher accuracy than the

Table 4.17. Metrics based on frames classifications of the best models on the validation dataset.

Model	Acc [%]	Precision [%]	Recall [%]	F1 [%]	Spec [%]
Logistic regression	92.52	79.94	18.26	29.74	99.56
SVM	92.83	90.91	19.23	31.74	99.82
Random forest	93.58	90.95	28.74	43.68	99.73
Gradient boosting	93.86	88.03	33.72	48.76	99.56
CRNN	95.81	82.24	65.95	73.20	98.65
RNN	95.89	83.42	66.25	73.85	98.74
Augmented CRNN	95.95	82.74	67.32	74.24	98.72
Augmented RNN	95.97	85.13	65.33	73.93	98.91
Semi-supervised CRNN	**96.09**	**84.49**	**67.22**	**74.87**	**98.83**
Semi-supervised RNN	**95.98**	**83.47**	**67.43**	**74.60**	**98.72**

Table 4.18. Metrics based on IOU using 50% threshold of the best models on the validation dataset.

Model	IOU Precision [%]	IOU Recall [%]	IOU F1 [%]
Logistic regression	25.61	8.66	12.94
SVM	32.42	11.34	16.80
Random forest	41.88	18.97	26.11
Gradient boosting	43.32	23.64	30.59
CRNN	73.78	45.84	56.55
RNN	81.06	47.35	59.78
Augmented CRNN	75.89	46.94	58.00
Augmented RNN	78.86	49.48	60.81
Semi-supervised CRNN	**78.14**	**49.14**	**60.34**
Semi-supervised RNN	**81.42**	**48.80**	**61.02**

RNN model which is probably related to the higher capability of the CRNN model to fit the bigger training data.

Table 4.19. Results of the best models on the test dataset with comparison to the result from engineering thesis.

Model	Acc [%]	Precision	Recall	F1 [%]	Spec [%]
CRNN1	93.41	85.16	68.59	75.98	97.86
CRNNFinal	**98.14**	**89.83**	**88.86**	**89.34**	**99.03**
RNNFinal	97.76	87.83	87.72	87.47	98.75

Figure 4.8 shows the IOU F1 score for the final CRNN and RNN models. Despite lower accuracy, the RNN model achieves a higher IOU F1 score, especially for higher thresholds. It is related to using a smaller 5ms frame, so the time intervals are more precise which is crucial for IOU based metrics.

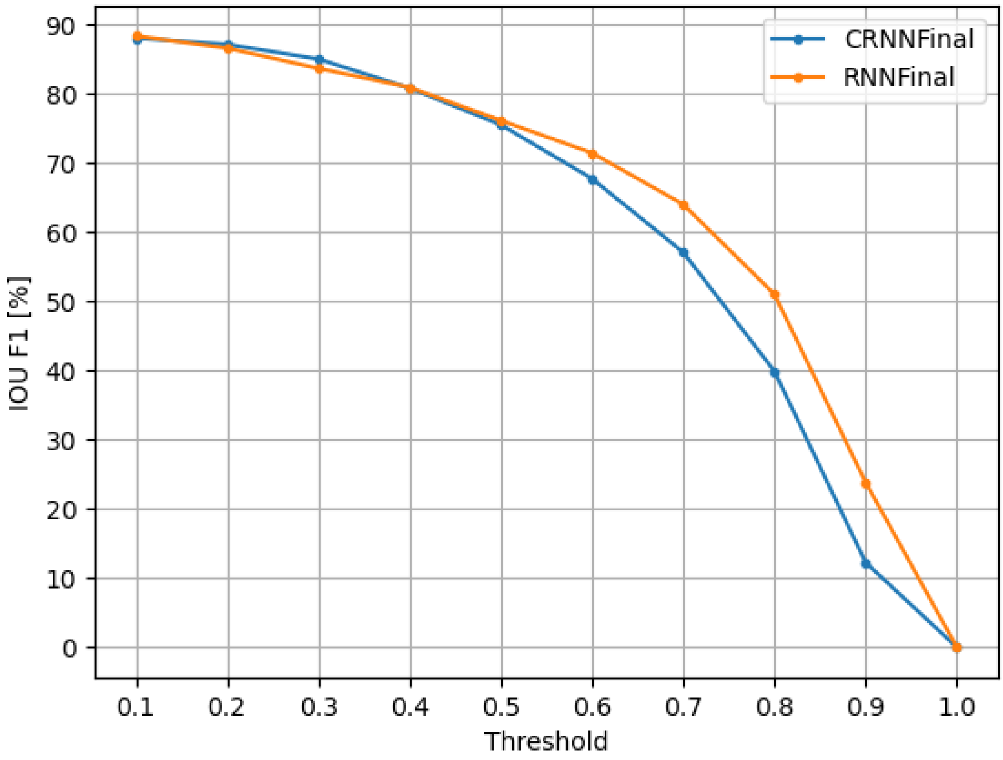


Figure 4.8. IOU F1 score of final CRNN and RNN models on the test dataset for a given threshold.

4.11. Performance metrics

All models were trained using an i7-12800H CPU with 32GB RAM and Ubuntu 22.04 64bit operating system. The versions of the libraries used for training the model are:

- Python=3.10.8
- NumPy=1.23.5
- pandas=1.5.2
- Librosa=0.9.2
- scikit-learn=1.2.0
- XGboost=1.7.4
- WandB=0.13.7
- TensorFlow=2.11.0

Training of the CRNNFinal model took 38 minutes and the RNNFinal model took 1 hour and 50 minutes. The inference time of 1 hour of recording takes approximately 12.21s for the CRNNFinal model and approximately 14.60s for the RNNFinal model. The inference times of both models are summarized in figure 4.9. RNNFinal model performance is worse mostly because it takes 400 long sequences as input and contains two GRU layers in comparison to the CRNNFinal model which takes only 200 long sequences as input and contains only one GRU layer.

The inference time could be optimized by adjusting batch size based on the used device. However, the bottleneck of the method is still the process of collecting bowel sound recordings from patients, so optimizing the inference time isn't crucial.

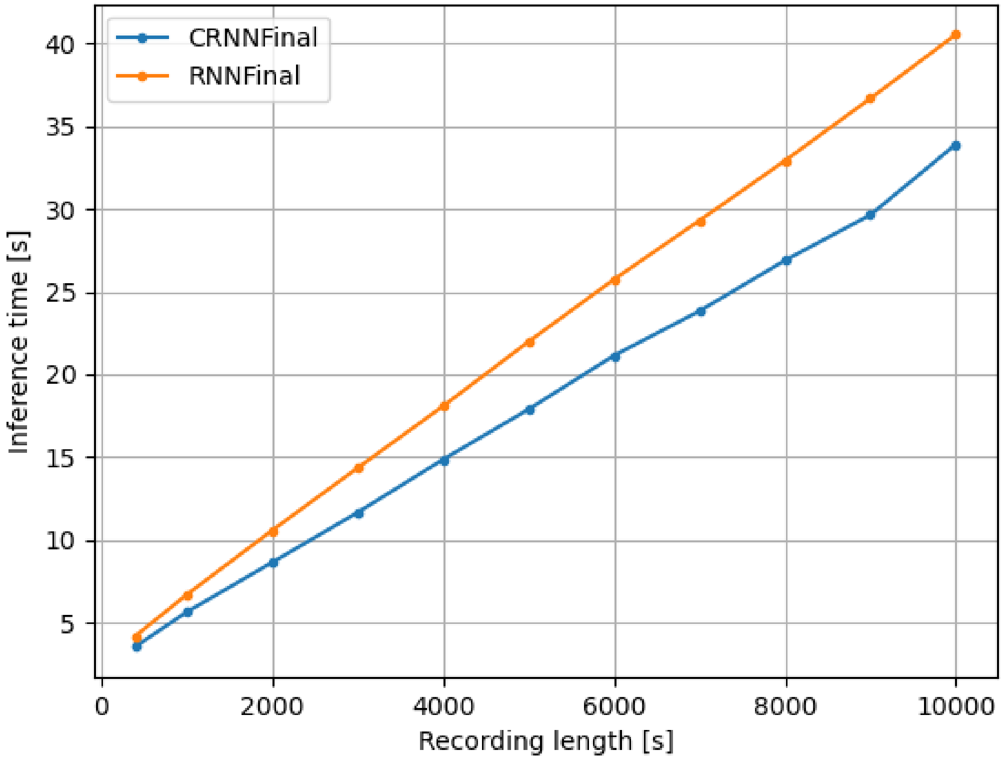


Figure 4.9. Inference time of final CRNN and RNN models based on the processed recording length.

5. Conclusions

In this thesis, the application for bowel sound analysis was built. The application allows users for training the models, calculating the quality of the models, visualizing the predictions, and generating bowel sound analysis reports. The bowel sound analysis tool was shared as a dockerized website and a packaged application.

The main research hypothesis was confirmed. Data augmentation and semi-supervised learning were able to significantly improve the used CRNN model. Additionally, a new sliding window method was proposed and tested using a series of machine learning methods but the acquired results were much worse than the sequence based methods. A new RNN model, with quality comparable to the CRNN model, was proposed and tested using smaller frame sizes, enabling more precise bowel sound localization. New IOU based metrics were implemented and used for additional model quality measurement.

The website that uses the CRNN model to generate reports based on the bowel sound recordings is served in `http://bowelsound.ii.pw.edu.pl`.

The work based on my engineering thesis and part of this master thesis was published in the article "Analysis of gastrointestinal acoustic activity using deep neural networks" [53].

The preprocessed bowel sound dataset is published in `www.kaggle.com/datasets/robertnowak/bowel-sounds` that can be used as a benchmark dataset for other researchers to compare the developed methods.

Processing the signal in the predicted time intervals of bowel sounds is an interesting path for future work. The achieved time intervals can be used to obtain segmented bowel sounds on spectrograms. It would enable including frequency and power analysis into the generated reports. Additionally, methods that make the time intervals of bowel sounds more precise than the frame width could be implemented using heuristic rules or adaptive filter approaches. It would allow for calculating duration based statistics more precisely.

In the case of popularizing bowel sound analysis as a clinical examination, integration of the sound analysis algorithm into the dedicated device can be useful. To do this, neural network compression like weight pruning and inference optimization should be added as an additional step of developing the method. Additionally, the bowel sound analysis website can be extended to store and manage uploaded recordings and analyses in the database and enable sharing of the results between doctors and researchers.

References

[1] J. F. Ficek, “Use of artificial neural networks to recognize bowel sounds”, Instytut Informatyki, 2021.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation”, *arXiv preprint arXiv:1406.1078*, 2014.

[3] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting”, *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[4] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization”, *arXiv preprint arXiv:1710.09412*, 2017.

[5] J. K. Nowak, R. Nowak, K. Radzikowski, I. Grulkowski, and J. Walkowiak, “Automated bowel sound analysis: An overview”, *Sensors*, vol. 21, no. 16, p. 5294, 2021.

[6] J. T. Farrar and F. J. Ingelfinger, “Gastrointestinal motility as revealed by study of abdominal sounds”, *Gastroenterology*, vol. 29, no. 5, pp. 789–802, 1955.

[7] C. Liatsos, L. J. Hadjileontiadis, C. Mavrogiannis, D. Patch, S. M. Panas, and A. K. Burroughs, “Bowel sounds analysis: A novel noninvasive method for diagnosis of small-volume ascites”, *Digestive diseases and sciences*, vol. 48, pp. 1630–1636, 2003.

[8] B. L. Craine, M. L. Silpa, and C. J. O’Toole, “Enterotachogram analysis to distinguish irritable bowel syndrome from crohn’s disease”, *Digestive diseases and sciences*, vol. 46, pp. 1974–1979, 2001.

[9] O. Sakata, Y. Suzuki, K. Matsuda, and T. Satake, “Temporal changes in occurrence frequency of bowel sounds both in fasting state and after eating”, *Journal of Artificial Organs*, vol. 16, pp. 83–90, 2013.

[10] K.-S. Kim, J.-H. Seo, and C.-G. Song, “Non-invasive algorithm for bowel motility estimation using a back-propagation neural network model of bowel sounds”, *Biomedical engineering online*, vol. 10, pp. 1–10, 2011.

[11] K. S. Kim, J. H. Seo, S. H. Ryu, M. H. Kim, and C. G. Song, “Estimation algorithm of the bowel motility based on regression analysis of the jitter and shimmer of bowel sounds”, *Computer methods and programs in biomedicine*, vol. 104, no. 3, pp. 426–434, 2011.

[12] K.-S. Kim, H.-J. Park, H. S. Kang, and C.-G. Song, “Awareness system for bowel motility estimation based on artificial neural network of bowel sounds”, in *4th International Conference on Awareness Science and Technology*, IEEE, 2012, pp. 185–188.

[13] U. D. Ulusar, M. Canpolat, M. Yaprak, S. Kazanir, and G. Ogunc, “Real-time monitoring for recovery of gastrointestinal tract motility detection after abdominal surgery”, in *2013 7th International Conference on Application of Information and Communication Technologies*, IEEE, 2013, pp. 1–4.

[14] U. D. Ulusar, “Recovery of gastrointestinal tract motility detection using naive bayesian and minimum statistics”, *Computers in biology and medicine*, vol. 51, pp. 223–228, 2014.

[15] K. Kölle, A. L. Fougner, R. Ellingsen, S. M. Carlsen, and Ø. Stavdahl, “Feasibility of early meal detection based on abdominal sound”, *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 7, pp. 1–12, 2019.

[16] M. A. Cheema, S. I. Siddiqui, and P. S. Rossi, “Comparison of different classifiers for early meal detection using abdominal sounds”, in *2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, IEEE, 2022, pp. 420–424.

[17] M. Li, J. Yang, and X. Wang, “Research on auto-identification method to the typical bowel sound signal”, in *2011 4th International Conference on Biomedical Engineering and Informatics (BMEI)*, IEEE, vol. 2, 2011, pp. 845–849.

[18] E. Bondareva, M. Constantinides, M. S. Eggleston, I. Jabłoński, C. Mascolo, Z. Radivojevic, and S. Šćepanović, “Stress inference from abdominal sounds using machine learning”, in *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, IEEE, 2022, pp. 1985–1988.

[19] G. Allwood, X. Du, K. M. Webberley, A. Osseiran, and B. J. Marshall, “Advances in acoustic signal processing techniques for enhanced bowel sound analysis”, *IEEE reviews in biomedical engineering*, vol. 12, pp. 240–253, 2018.

[20] X. Du, G. Allwood, K. M. Webberley, A. Osseiran, and B. J. Marshall, “Bowel sounds identification and migrating motor complex detection with low-cost piezoelectric acoustic sensing device”, *Sensors*, vol. 18, no. 12, p. 4240, 2018.

[21] X. Du, G. Allwood, K. M. Webberley, A. Osseiran, W. Wan, A. Volikova, and B. J. Marshall, “A mathematical model of bowel sound generation”, *the Journal of the Acoustical Society of America*, vol. 144, no. 6, EL485–EL491, 2018.

[22] Z. Chen and W. Montlouis, “Bowel movement signal modeling and parameters extraction”, in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2020, pp. 963–967.

[23] C. Dimoulas, G. Kalliris, G. Papanikolaou, and A. Kalampakas, “Novel wavelet domain wiener filtering de-noising techniques: Application to bowel sounds captured by means of abdominal surface vibrations”, *Biomedical signal processing and control*, vol. 1, no. 3, pp. 177–218, 2006.

[24] ——, “Long-term signal detection, segmentation and summarization using wavelets and fractal dimension: A bioacoustics application in gastrointestinal-motility monitoring”, *Computers in Biology and Medicine*, vol. 37, no. 4, pp. 438–462, 2007.

[25] C. Dimoulas, G. Kalliris, G. Papanikolaou, V. Petridis, and A. Kalampakas, “Bowel-sound pattern analysis using wavelets and neural networks with application to long-term, unsupervised, gastrointestinal motility monitoring”, *Expert Systems with Applications*, vol. 34, no. 1, pp. 26–41, 2008.

[26] C. A. Dimoulas, G. Papanikolaou, and V. Petridis, “Pattern classification and audiovisual content management techniques using hybrid expert systems: A video-assisted

bioacoustics application in abdominal sounds pattern analysis", *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 082–13 093, 2011.

[27] C.-F. Tsai, T.-J. Wu, and Y.-M. Chao, "Labview based bowel-sounds monitoring system in realtime", in *2011 International Conference on Machine Learning and Cybernetics*, IEEE, vol. 4, 2011, pp. 1815–1818.

[28] T. Emoto, K. Shono, U. R. Abeyratne, T. Okahisa, H. Yano, M. Akutagawa, S. Konaka, and Y. Kinouchi, "Arma-based spectral bandwidth for evaluation of bowel motility by the analysis of bowel sounds", *Physiological Measurement*, vol. 34, no. 8, p. 925, 2013.

[29] M.-J. Sheu, P.-Y. Lin, J.-Y. Chen, C.-C. Lee, and B.-S. Lin, "Higher-order-statistics-based fractal dimension for noisy bowel sound detection", *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 789–793, 2014.

[30] K. Kölle, M. F. Aftab, L. E. Andersson, A. L. Fougner, and Ø. Stavdahl, "Data driven filtering of bowel sounds using multivariate empirical mode decomposition", *BioMedical Engineering OnLine*, vol. 18, pp. 1–20, 2019.

[31] Y. Ogino, Y. Satoh, and O. Sakata, "Forecasting bowel sound occurrence frequency by sarima model", in *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, IEEE, 2019, pp. 219–223.

[32] Y. Yamada, O. Sakata, and Y. Satoh, "Hybrid bowel sound measurement system combining microphones and a vibration sensor", in *Proceedings of the 12th International Conference on Computer Modeling and Simulation*, 2020, pp. 175–179.

[33] K. Kodani and O. Sakata, "Automatic bowel sound detection under cloth rubbing noise", in *2020 IEEE REGION 10 CONFERENCE (TENCON)*, IEEE, 2020, pp. 779–784.

[34] S. Rajkumar, K. Sathesh, and N. K. Goyal, "Neural network-based design and evaluation of performance metrics using adaptive line enhancer with adaptive algorithms for auscultation analysis", *Neural Computing and Applications*, vol. 32, pp. 15 131–15 153, 2020.

[35] B.-S. Lin, M.-J. Sheu, C.-C. Chuang, K.-C. Tseng, and J.-Y. Chen, "Enhancing bowel sounds by using a higher order statistics-based radial basis function network", *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 675–680, 2013.

[36] Y. Yin, W. Yang, H. Jiang, and Z. Wang, "Bowel sound based digestion state recognition using artificial neural network", in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, 2015, pp. 1–4.

[37] Y. Yin, H. Jiang, W. Yang, and Z. Wang, "Intestinal motility assessment based on legendre fitting of logarithmic bowel sound spectrum", *Electronics Letters*, vol. 52, no. 16, pp. 1364–1366, 2016.

[38] Y. Yin, H. Jiang, S. Feng, J. Liu, P. Chen, B. Zhu, and Z. Wang, "Bowel sound recognition using svm classification in a wearable health monitoring system", *Science China Information Sciences*, vol. 61, pp. 1–3, 2018.

[39] R. Sato, T. Emoto, Y. Gojima, and M. Akutagawa, “Automatic bowel motility evaluation technique for noncontact sound recordings”, *Applied Sciences*, vol. 8, no. 6, p. 999, 2018.

[40] K. Horiyama, T. Emoto, T. Haraguchi, T. Uebanso, Y. Naito, T. Gyobu, K. Kanemoto, J. Inobe, A. Sano, M. Akutagawa, *et al.*, “Bowel sound-based features to investigate the effect of coffee and soda on gastrointestinal motility”, *Biomedical Signal Processing and Control*, vol. 66, p. 102 425, 2021.

[41] J. Liu, Y. Yin, H. Jiang, H. Kan, Z. Zhang, P. Chen, B. Zhu, and Z. Wang, “Bowel sound detection based on mfcc feature and lstm neural network”, in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, 2018, pp. 1–4.

[42] F. Wang, D. Wu, P. Jin, Y. Zhang, Y. Yang, Y. Ma, A. Yang, J. Fu, and X. Feng, “A flexible skin-mounted wireless acoustic device for bowel sounds monitoring and evaluation”, *Science China Information Sciences*, vol. 62, pp. 1–11, 2019.

[43] G. Wang, Y. Yang, S. Chen, J. Fu, D. Wu, A. Yang, Y. Ma, and X. Feng, “Flexible dual-channel digital auscultation patch with active noise reduction for bowel sound monitoring and application”, *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 7, pp. 2951–2962, 2022.

[44] N. Wang, A. Testa, and B. J. Marshall, “Development of a bowel sound detector adapted to demonstrate the effect of food intake”, *BioMedical Engineering OnLine*, vol. 21, no. 1, pp. 1–12, 2022.

[45] X. Zheng, C. Zhang, P. Chen, K. Zhao, H. Jiang, Z. Jiang, H. Pan, Z. Wang, and W. Jia, “A crnn system for sound event detection based on gastrointestinal sound dataset collected by wearable auscultation devices”, *IEEE Access*, vol. 8, pp. 157 892–157 905, 2020.

[46] K. Zhao, H. Jiang, T. Yuan, C. Zhang, W. Jia, and Z. Wang, “A cnn based human bowel sound segment recognition algorithm with reduced computation complexity for wearable healthcare system”, in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2020, pp. 1–5.

[47] K. Zhao, H. Jiang, Z. Wang, P. Chen, B. Zhu, and X. Duan, “Long-term bowel sound monitoring and segmentation by wearable devices and convolutional neural networks”, *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 5, pp. 985–996, 2020.

[48] K. Zhao, S. Feng, H. Jiang, Z. Wang, P. Chen, B. Zhu, and X. Duan, “Wearable bowel sound monitoring with quality enhancement using u-net”, in *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, IEEE, 2022, pp. 2443–2447.

[49] C. Sitaula, J. He, A. Priyadarshi, M. Tracy, O. Kavehei, M. Hinder, A. Withana, A. McEwan, and F. Marzbanrad, “Neonatal bowel sound detection using convolutional neural network and laplace hidden semi-markov model”, *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1853–1864, 2022.

[50] L. Burne, C. Sitaula, A. Priyadarshi, M. Tracy, O. Kavehei, M. Hinder, A. Withana, A. McEwan, and F. Marzbanrad, “Ensemble approach on deep and handcrafted

features for neonatal bowel sound detection", *IEEE Journal of Biomedical and Health Informatics*, 2022.

[51] P. Zhou, M. Lu, P. Chen, D. Wang, Z. Jin, and L. Zhang, "Feasibility and basic acoustic characteristics of intelligent long-term bowel sound analysis in term neonates.", *Frontiers in Pediatrics*, vol. 10, pp. 1 000 395–1 000 395, 2022.

[52] Y. Kutsumi, N. Kanegawa, M. Zeida, H. Matsubara, and N. Murayama, "Automated bowel sound and motility analysis with cnn using a smartphone", *Sensors*, vol. 23, no. 1, p. 407, 2023.

[53] J. Ficek, K. Radzikowski, J. K. Nowak, O. Yoshie, J. Walkowiak, and R. Nowak, "Analysis of gastrointestinal acoustic activity using deep neural networks", *Sensors*, vol. 21, no. 22, p. 7602, 2021.

[54] F. Shaffer and J. P. Ginsberg, "An overview of heart rate variability metrics and norms", *Frontiers in public health*, p. 258, 2017.

[55] C. Yan, P. Li, L. Ji, L. Yao, C. Karmakar, and C. Liu, "Area asymmetry of heart rate variability signal", *Biomedical engineering online*, vol. 16, pp. 1–14, 2017.

[56] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[57] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition", *arXiv preprint arXiv:1904.08779*, 2019.

[58] T. Inoue, P. Vinayavekhin, S. Wang, D. Wood, A. Munawar, B. J. Ko, N. Greco, and R. Tachibana, "Shuffling and mixing data augmentation for environmental sound classification", 2019.

[59] P. Cascante-Bonilla, F. Tan, Y. Qi, and V. Ordonez, "Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 6912–6920.

[60] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence", *Advances in neural information processing systems*, vol. 33, pp. 596–608, 2020.

[61] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space", in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

List of Figures

List of Tables