

**BABEȘ-BOLYAI UNIVERSITY CLUJ-NAPOCA**  
**FACULTY OF MATHEMATICS AND COMPUTER**  
**SCIENCE**  
**SPECIALIZATION COMPUTER SCIENCE**

## **DIPLOMA THESIS**

**Predicting watermelon ripeness based  
on acoustic signals using machine  
learning techniques**

**Supervisor**  
**Lector Dr. Ioan Gabriel MIRCEA**

*Author*  
*Petruț Adrian Damian*

2021



---

## ABSTRACT

---

Ripeness is an important factor in determining watermelon quality. Watermelons do not continue to ripen post harvest, therefore watermelons harvested at the wrong stage that reach consumers will not be of good eating quality. This paper investigates intelligent methods to predict watermelon ripeness based on acoustic signals recorded using mobile devices.

*The main contribution of this paper is a proposed method to select the most relevant features of the acoustic signal, using a genetic algorithm. The selected features are used to train a neural network classification model. Our experiments show that the proposed approach results in better performance and robustness of neural networks, with a mean accuracy of 96% using cross validation evaluation.*

This work is the result of my own activity. I have not received nor given unauthorized help.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related work</b>	<b>2</b>
2.1	Watermelon quality indicators . . . . .	2
2.2	Non-destructive methods of evaluation . . . . .	2
2.2.1	Electrical and magnetic technologies . . . . .	2
2.2.2	Near infrared spectroscopy . . . . .	3
2.2.3	Machine vision . . . . .	4
2.2.4	Acoustic technology . . . . .	5
2.2.5	Other methods . . . . .	8
2.3	Advantages and disadvantages . . . . .	8
<b>3</b>	<b>Proposed approach</b>	<b>9</b>
3.1	Data harvesting . . . . .	9
3.2	Preprocessing . . . . .	10
3.3	Feature Extraction . . . . .	12
3.3.1	Short-time energy . . . . .	12
3.3.2	Zero crossing rate . . . . .	13
3.3.3	Frequency bands energy ratio . . . . .	13
3.4	Experiments and Results . . . . .	14
3.4.1	ANN training . . . . .	15
3.4.2	Genetic algorithm - Feature selection . . . . .	19
3.4.3	GA results . . . . .	20
3.4.4	Final model . . . . .	23
<b>4</b>	<b>Software Implementation</b>	<b>25</b>
4.1	Software design and architecture . . . . .	25
4.1.1	Use case diagram . . . . .	26
4.1.2	Component diagram . . . . .	27
4.1.3	Sequence diagram . . . . .	27
4.2	Technologies used . . . . .	29

4.2.1	Ionic React . . . . .	29
4.2.2	Koa . . . . .	29
4.2.3	Database storage . . . . .	29
4.2.4	Python . . . . .	29
4.3	Testing & Specifications . . . . .	29
4.3.1	API specifications . . . . .	29
4.3.2	Testing . . . . .	30
4.4	UserManual . . . . .	31
4.4.1	Login & Register . . . . .	31
4.4.2	Home screen . . . . .	32
4.4.3	Submit Recording . . . . .	32
4.4.4	View recordings history . . . . .	34
<b>5</b>	<b>Conclusions</b>	<b>35</b>
5.1	Personal contributions . . . . .	35
5.2	Further work . . . . .	35

# Chapter 1

## Introduction

The watermelon(*Citrullus lanatus*) is a tropical fruit belonging to the Cucurbitaceae family. It is one of the most popular fruits, with increasing demand especially during the summer season. China, the world's largest watermelon producer, experienced an average annual production increase of +1.9% from 2007 to 2018 according to The Food and Agriculture Organization (FAO) statistics.

Watermelons do not continue to ripen post harvest, therefore it is important for average consumers to be able to identify and avoid bad quality watermelons. Watermelon quality inspection is traditionally performed by checking characteristics such as external color, peel texture, sound response when thumped ( Marita 1996). This approach requires experience and is also prone to errors, thus it is not suitable for average consumers who shop at the supermarket.

Due to the inefficiency of traditional methods of evaluating watermelon quality, there has been a lot of research regarding automatic non-destructive evaluation. The aim of this paper is to provide a practical implementation of an automatic quality detection system for watermelons suitable for average consumers.

The present paper focuses on the detection of watermelon ripeness based on acoustic signals. We proposed a method that uses a genetic algorithm to select relevant features from the acoustic signal which are used to train an artificial neural network.

# Chapter 2

## Related work

This chapter discusses existing approaches for solving the mentioned problem that are documented in the scientific literature. We discuss advantages and disadvantages of existing methods, and areas to improve.

### 2.1 Watermelon quality indicators

There are various factors that affect the overall internal quality of watermelons: internal texture (higher quality watermelons have firm flesh/pulp) is typically assessed using an MT (Magness-Taylor) penetration test which has the disadvantage of being a destructive method of evaluation, damaging the tissue; sugar content can be estimated by measuring the soluble solids content(SSC) in watermelons as the sugars comprise around 80% or more of the soluble solids content, ripe watermelon should have between 10% to 14% soluble solids content Marita 1996; hollow heart is considered an internal defect that consists of a void/cavity in the flesh of the watermelon.

### 2.2 Non-destructive methods of evaluation

Due to the disadvantages of destructive methods of assessment, there has been extensive research into the development of non-destructive methods of evaluating the internal quality of watermelons. An overview of different approaches for non-destructive quality detection is detailed below.

#### 2.2.1 Electrical and magnetic technologies

These methods use electrical instruments to measure indices correlated to watermelon quality. Ripe watermelons usually have higher density and watermelons

with internal defects such as hollow heart have low density. Kato and Matuda 2004 have researched the relationship between density, transmitted light through watermelons and firmness, and found that SSC could be approximated using multiple regression analysis. They stated that higher density watermelons had higher SSC values and no cavities.

Magnetic resonance imaging(MRI) can be used for non-destructive testing for internal defects in watermelons. Saito et al. 1996 used an MRI for detection of cavities in watermelons. The results were accurate 28 times out of the 30 samples tested. MRI testing is, however, expensive and is not able to detect other quality related aspects such as ripeness and firmness.

S. O. Nelson and Kays 2007 investigated the correlation between internal and external dielectric properties of watermelons and the soluble solids content. An open-ended coaxial-line probe was used to measure dielectric properties of both internal and external tissue. Their findings show that individual dielectric properties of both the external and internal have a low correlation with SSC. This study required special instruments to be able to make the measurements necessary and the authors stated that further research is required to assess the viability of practical and useful techniques that could be used reliably to predict eating quality based on dielectric properties of watermelons.

### 2.2.2 Near infrared spectroscopy

Near Infrared radiation is a type of radiation that is outside the visible spectrum. When a beam of light falls on an object, the object can reflect absorb or transmit the light. It is possible to study an object's chemical composition by examining the relative proportions of these 3 phenomena. There is research showing near infrared spectroscopy can be used to approximate SSC in watermelons.

Dull, Birth, et al. 1989 tested both intact and sliced cantaloupe melons and used two wavelengths to predict SSC. The results were more accurate using the sliced melons obtaining  $r = -0.97$  and  $r = -0.6$  for the sliced and intact melons respectively; the standard error of calibration (SEC) and standard error of prediction (SEP) were  $SEC = 0.56$ ,  $SEP = 1.56$  and  $SEC = 1.67$ ,  $SEP = 2.18$  for the sliced and intact melons respectively. Dull, Birth, et al. 1990 used a NIR spectrophotometric method for approximating SSC in honeydew melons. They used a 45 degrees angle between the detector and the source incident beam and using regression analysis of the spectra and chemical data that utilises a ratio of two second derivatives they developed a model for predicting SSC with  $r = 0.85$  and  $SEC = 1.5$ . Later Dull, Leffler, et al. 1992 made an instrument to evaluate SSC of honeydew melons;  $r = 0.91$ ,  $SEC = 0.82\%$ ,  $SEP = 1.85\%$ .



Ito, Morimoto, et al. 2002 evaluated SSC non-destructively using a multiple regression equation based on the wavelengths of 902 nm and 872 nm; the results were correlation coefficient  $r = 0.87$ , root mean square error of calibration (RMSEC) 0.56% and root mean square error of prediction (RMSEP) 0.664%. Abebe 2006 analysed the diffuse transmittance spectra and selected four wavelengths (770 nm, 830 nm, 865 nm, and 887 nm) and developed a calibration model to predict SSC with  $r = 0.81$  and SEP = 0.42%. Even with a low-cost spectrometer, Tian, Ying, Xu, et al. 2009 developed a model to predict SSC with high degree of correlation ( $r = 0.89$ , RMSEC = 0.494, RMSEP = 0.665) using visible/near infrared (Vis/NIR) diffuse transmittance method. Tian, Ying, Lu, et al. 2007 also used the Vis/NIR technique to predict firmness and discovered that a model using partial least squares regression (PLS) gave the best results:  $r = 0.892$ , RMSEP = 0.589.

Flores et al. 2008 assessed the total soluble solids content of intact and cut watermelons using NIR technology (NIR diode array spectrometer). The results were more accurate for cut watermelons :  $r = 0.92$  and  $r = 0.81$  for cut and intact watermelons respectively. D. Jie and Ying 2014 used Vis/NIR technology to develop a non-destructive on-line detection prototype for SSC of watermelons. Multiple models were tested, the best one obtained  $r = 0.66$  and RMSEP = 0.39 in on-line testing of samples. The authors noted that the model parameters would need to be optimised further but showed the potential of using such detection system for on-line testing of watermelons.

J. Guthrie and K. Walsh 1998 developed a calibration model using NIR spectroscopy to assess total soluble solids content in watermelons base on a wave-length range of 700-1100 nm. He also studied the robustness of such models. When testing the models on populations of watermelons that were harvested in different periods he discovered that the calibration models were lacking robustness.

C. V. Greensill and K. B. Walsh 2001 Greensill studied calibration transfer within different spectrometers to measure SSC using several different methods. He compared standardisation and model updating methods. He also noted that robustness of the calibration model when testing watermelon batches that were harvested at different times could be improved when applying a wavelet transform procedure of the spectra.

### 2.2.3 Machine vision

Machine vision system refer to the application of computer automated processing techniques on image sensor data in order to solve various problems. Traditional methods of evaluation of watermelon quality also include assessment of external appearance such as color and texture. A machine vision system could potentially

allow automatic and non-destructive evaluation of watermelon quality based on external appearance.

Koc [2007](#) conducted a study on watermelon volume using image processing techniques and compared the results with measurements using water displacement and ellipsoid approximation. The images were captured using a CMOS camera with a white cardboard as background for the watermelons and two pictures were taken on a 0 degrees and 90 degrees angle around the longitudinal axis. The results showed that the image processing method obtained values closer to the ones obtained from water displacement than the value obtained using ellipsoid approximation. The average difference in volume estimation using the image processing method was 7.7%.

Machine learning methods are commonly used for image analysis tasks such as image recognition/pattern recognition, and can be used to detect complex patterns with high accuracy. S. R. M. Shah Baki and Shazana [2009](#) developed an artificial neural network (ANN) model to detect ripeness. Images were captured with a digital camera and pre-processed from RGB color scale to YCbCr scale from which the features for the neural network were computed. The model achieved an accuracy of 86.5% with an error lower than 1.1%. N. A. Syazwan and Nooritawati [2012](#) studied what is the best capture position for evaluating watermelon maturity. He extracted features from the RGB scale representation and tested five different positions. Results showed the angle capturing the yellow spot on the watermelon was the best position obtaining an accuracy of 73% while the bottom position obtained an accuracy of 20%.

### **2.2.4 Acoustic technology**

When a sound wave reaches an object the transmitted or reflected wave depends on the acoustic characteristics of the object. When it comes to agricultural products such as watermelon, this can reveal information about the internal structure of the melon. Acoustic evaluation is based on the hypothesis that bad quality watermelons yield a different acoustic response compared to good quality ones. Traditionally, acoustic response is also used by human experts when judging the quality of the watermelon. The sound made when the watermelon is thumped on the sides by hand is different between ripe and unripe watermelons, however this technique requires experience so only human experts can judge ripeness using this method.

J. Sugiyama and Usui [1994](#) analysed the acoustic response of muskmelons as result of the impact with a wooden ball pendulum at 45 degrees angle. They showed that the impulse waveform created by the impact was transmitted with uniform velocity. Their findings also suggest that transmission velocity decreases as the mel-

ons become ripe. They developed an instrument to measure transmission velocity and tested its performance against a group of human judges that rated melons' firmness on a scale of one to three. The correlation coefficient obtained with the instrument was  $r = 0.832$ . Later Sugiyama et al. 1998 developed a portable firmness tester; correlation coefficient between apparent elasticity and transmission velocity was  $r = 0.943$ . This tester could be used to track internal changes in ripening melons. N. Ozer and Simon 1998 used a multiple impact approach in order to diminish the variability of acoustic response depending on the angle and location of impact. They tested cantaloupe melons and obtained a correlation coefficient  $r = 0.86$  for flesh firmness and  $r = 0.94$  for elastic modulus.

During commercial handling of watermelons it is common for accidental drops to occur, damaging the watermelons. Depending on the scenario watermelons can suffer internal damage without being externally visible. P. R. Armstrong and Brusewitz 1997 analysed the changes in acoustic response after dropping the watermelons successively from a height reasonably possible in commercial handling. He took compression measurements and recorded acoustic characteristics before and after each successive drop. The results showed a significant difference in compression related measurements after each drop, however using acoustic parameters only large differences in internal damage could be detected.

Ito and Sugiyama 2002 used a portable firmness tester to measure transmission velocity in order to judge the right time for harvesting watermelons. He also found that the velocity of the impulse wave was slower as watermelons ripened and the sucrose content was higher than 8.5% when transmission velocity was lower than 80 m/s. B. Diezema and Orihuel 2003 constructed a device composed of a microphone, structural components and a mechanical impact generator. Based on the acoustic response he detected hollow heart defect in seedless watermelons. They chose band magnitude parameter over resonant frequencies parameter as it had a higher correlation of  $r = 0.62$ - $0.67$ . The final results recorded an accuracy of 78.14% for detection of hollow watermelons.

Lee et al. 2006 developed a model to predict internal cavities based on acoustic response. Watermelons were hit with a constant-force hitting hammer and the acoustic response was recorded using 3 sensors. The watermelons were visually rated into one of 6 categories depending on the size of the internal cavity. The prediction model was developed using Partial Least Square Regression and overall accuracy obtained was 90.1%.

N. Jamal and Rao 2005 looked at the relationship between firmness and resonant frequency. A wooden pendulum ball was used to excite the watermelon. The correlation between resonant frequency and Young's modulus was 0.95. It was noted that the angle and location of the impact had a significant influence on the acoustic

response and the center part of the fruit surface was suggested.

Baki et al. 2010 derived mel-frequency cepstrum coefficients(MFCC) from the acoustic signal. These were used as features to train a multi-layer perceptron neural network to classify ripe and unripe watermelons and obtained an accuracy of 77.25%. J. Mao and Wang 2016 checked the effects of fruit tray and hitting ball material on the resonant frequency and found that a plastic tray would result in a smaller standard deviation of resonant frequency than a rubber tray; a stainless steel hitting ball also resulted in more consistent resonant frequency across samples, however the difference was not a major. A correlation was observed between fruit firmness and firmness indices such as  $f^2m$ ,  $f^2m^{2/3}$  and acoustic parameters such as  $MI1$ ,  $MI1$  (index of the first/second moment). A prediction model using ANN was developed. Correlation of  $MI1$  index using ANN was  $r = 0.739$  and  $r = 0.684$  for calibration and validation set respectively. It was noted that compared to a linear model, ANN model only improved correlation slightly suggesting the relationship between  $MI1$  and firmness is closer to linearity.

Y. Wei and Qi 2012 developed a device to predict sugar content in watermelons. Six frequencies were chosen and a Stepwise Multiple Linear Regression (SMLR) model was developed based on acoustic transmissivity at those frequencies. The best results were obtained when tapping the watermelon at the equatorial position with  $r = 0.807$ .

W. Zeng and Mcloughlin 2013 captured acoustic signals when tapping a watermelon using mobile devices. The impact frames were segmented and acoustic characteristics were extracted and analysed. Each watermelon was labeled as ripe or unripe and the support vector machine method was used to develop a classification model. The model was trained using the extracted features individually and in combination. Findings showed that brightness feature was not able to differentiate between ripe/unripe melons resulting in an overall accuracy of 47.5%. Examining the frequency-domain transformed signal, it was observed that energy of acoustic response was mostly concentrated in a low sub-band frequency in unripe watermelons. The overall accuracy using the sub-band energy ratio feature was 82.7%. The combined feature vector resulted in an overall accuracy of 89.3%.

X. Chen and Deng 2018 employed a novel technique based on wavelet multi-resolution decomposition of the acoustic signal. A five level wavelet decomposition resulted in the best discrimination index. It was also found that out of 4137 wavelet coefficients, 485 of them were normal random variables for both ripe and unripe watermelons. Based on the best wavelet coefficient, an inequality for detecting ripeness was derived from a statistical hypothesis test. The method was effecting, and the final testing accuracy obtained was 91.76%.

### 2.2.5 Other methods

R. Abbaszadeh and Najafi 2015 collected vibration data from watermelons using a vibration plate and shaker and response was optically recorded using a Laser Doppler vibrometer. The signals were transformed from time-domain to frequency-domain by Fast Fourier transform and twenty-nine features were extracted. K-nearest neighbor method was used to train a model for classification into ripe/unripe categories. Highest accuracy was obtained for K=1 and K=2 with 94.73% accuracy for a test set of 19 samples.

## 2.3 Advantages and disadvantages

There are various approaches when it comes to non-destructive evaluation of watermelon quality as showed above. Most of these approaches have been able to achieve good results, however there are differences in costs associated and accuracy of detection. Electrical and magnetic methods discussed can detect internal defects very well but are usually expensive as require special equipment such as MRI machine. NIR technology can achieve accurate results and detect internal qualities such as SSC, firmness and hollow heart. NIR technology also requires less expensive equipment and could be effective and practical for on-line detection in industrial applications. Acoustic methods are simpler and inexpensive and more recent methods showed they can still achieve high accuracy of detection. The mentioned approaches are applicable for industrial use, however the aim of this current paper is to develop a practical and easy to use system suitable for average consumers, that require no special equipment but can still reliably predict watermelon quality; taking this into consideration, an approach based on acoustic detection seems to be best suited as it is simple to implement and also low-cost.

# Chapter 3

## Proposed approach

The aim of this paper is to design and implement a classification model that can be used to predict watermelon ripeness based on a sound recording taken with a mobile device such as a smartphone. A recording containing three watermelon knocking events is considered. The knocking events will be segmented and combined into a single signal. The signal is processed and frequency domain features are extracted. Using a genetic algorithm the most relevant features are selected. Finally a classification model is built using neural networks. This chapter will discuss in detail the proposed approach and the results obtained in our experiments.

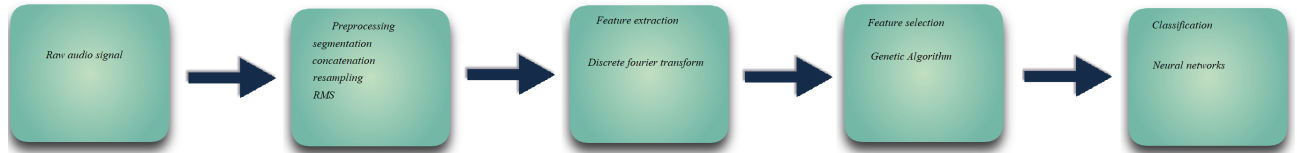


Figure 3.1: Proposed approach - general diagram

### 3.1 Data harvesting

A total of 104 watermelons were gathered from local supermarkets. A recording was taken from each sample by knocking on the watermelon 3 times around the equatorial region. All the recordings were taken in .wav format with a sample rate of 44100Hz. After a recording was taken each watermelon was cut and labelled as ripe or unripe based on manual inspection and taste test. Out of 104 samples 53 were labelled as ripe and 51 were labelled as unripe.

## 3.2 Preprocessing

Each recording contains three separate knocking events. Before analyzing the data, the three events need to be segmented and concatenated into a single signal. For this task a python script was written. First, the entire signal is separated into mini frames each comprised of  $p$  consecutive signal values. For each mini frame the root mean square is calculated using the following formula:

$$RMS = \sqrt{\frac{1}{n} \sum_i^n s_i^2}$$

n - number of samples in window  
s - initial signal

Following this, the idea is to compare the RMS values of a series of adjacent mini frames to determine when the thumping event starts. There should be a significant increase in the frames at the start of the knocking compared to the frames just before it. The length  $p$  of a mini frame needs to be small enough to detect the start of an event early, however large enough so that it's not too sensitive to small variations before the actual event signal. Visually analyzing the audio signals using the Audacity software,  $p = 40$  was chosen for the mini frame size and used overlapping windows at the halfway point.

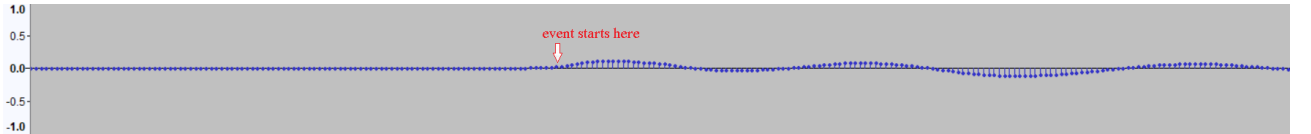


Figure 3.2: Audacity software view of the start of a knocking event signal

The start frame of an event will be identified by comparing the average RMS value of a window of mini-frames before and after the start frame. Formally we will identify a mini frame at index  $k$  as the frame signaling the start of a knocking event if the following criteria is met:

$$\frac{F}{m} \sum_{i=1}^m r_{k+i} > \frac{1}{m} \sum_{i=1}^m r_{k-i}$$

$r_i$  - RMS of mini frame at index  $i$   
F - increase factor  
m - length of window of adjacent mini frames to consider

The parameter  $F$  specifies how much of an increase the frames at the start of the event should have compared to the frames before it. The parameter  $m$  specifies how many mini frames should be used to compute the average RMS of the 2 windows before and after the start frame.  $F$  and  $m$  were experimentally determined for our dataset to be 9 and 4 respectively. After identifying the start frame, the signal will be segmented using a fixed length of 3600 samples. Three knocking events are identified and concatenated which will then be used for further analyzing. Below is a waveform graphic comparison of the signal before and after segmentation&concatenation.

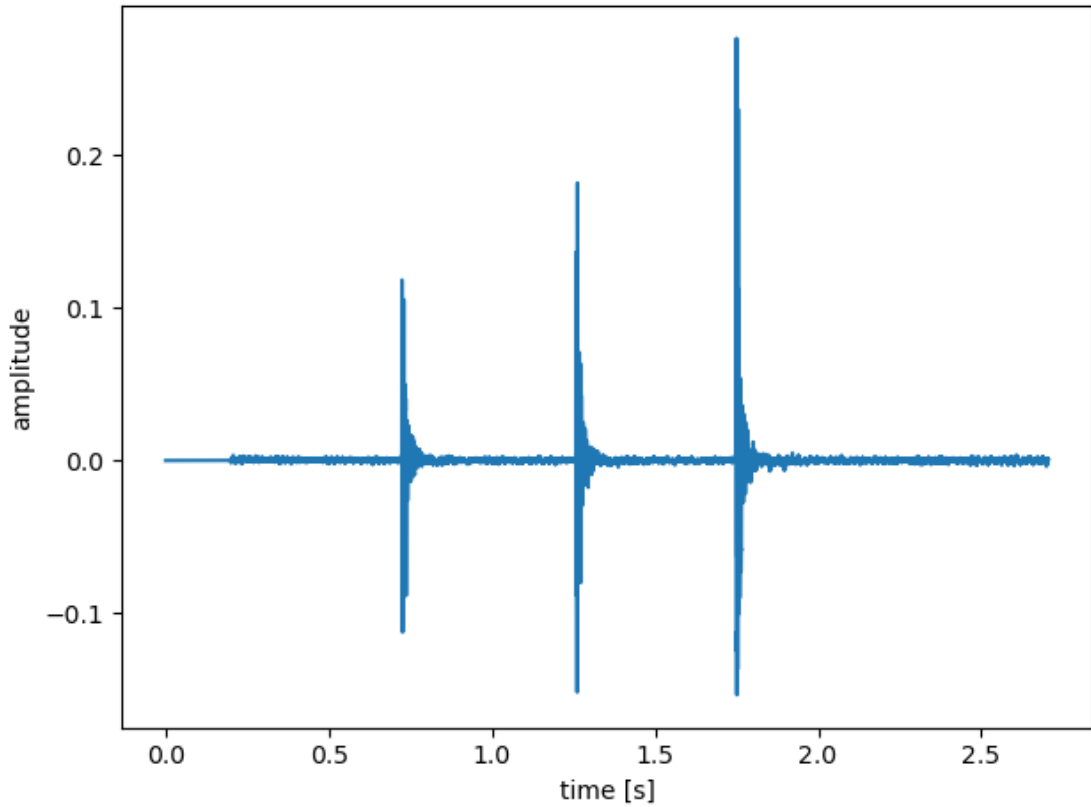


Figure 3.3: Initial signal



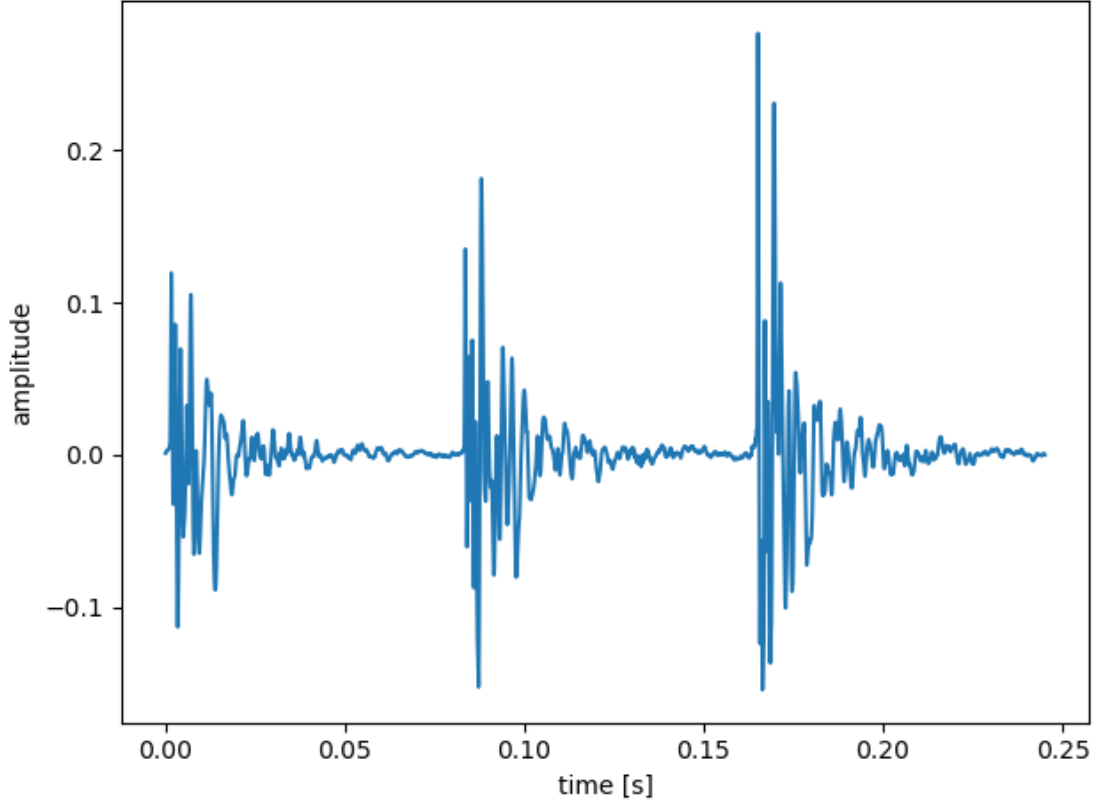


Figure 3.4: Processed signal

### 3.3 Feature Extraction

In order to find discrimination factors based on the recorded signals (post processing) it is necessary to extract features or characteristics of the sound which will then be further analyzed. W. Zeng and Mcloughlin [2013](#) used a support vector machine to predict watermelon ripeness based on sound recordings taken with a mobile device. The results showed that features such as short-time energy (STE), zero crossing rate (ZCR) and frequency domain derived characteristics were relevant in discriminating between ripe and unripe watermelons.

#### 3.3.1 Short-time energy

The short-time energy features represents the power of the entire signal and is calculated by computing the sum of the squared amplitude values of the signal in time domain.

$$STE = \sum_{i=1}^n s_i^2$$

$s_i$  - amplitude value at index i of the signal  
 $n$  - length of signal

### 3.3.2 Zero crossing rate

Zero crossing rate (ZCR) measures the number of time domain zero crossings within a signal and is calculated using the formula below.

$$ZCR = \frac{1}{2n} \sum_{i=1}^n | \text{sign}(s_i) - \text{sign}(s_{i-1}) |$$

$s_i$  - amplitude value at index i of the signal  
 $n$  - length of signal  
 $\text{sign} - \text{sign}(x) = \begin{cases} 1 & 0 \leq x \\ -1 & x < 0 \end{cases}$

### 3.3.3 Frequency bands energy ratio

The previously mentioned features are computed based on the time domain representation of a signal. The discrete Fourier Transform (DFT) is a commonly used method in sound analysis. The DFT can be used to convert the signal from a time domain representation to a frequency domain representation using the formula below: The first element  $y[0]$  is called the DC term and is usually discarded as it is the

$$y[k] = \sum_{n=0}^{N-1} e^{-2i\pi \cdot \frac{n \cdot k}{N}} \cdot x[n], k = 0, 1, \dots, N - 1$$

$x[n]$  - amplitude value at index n of the signal  
 $N$  - length of signal

sum of the initial signal samples. Half of the DFT output corresponds to positive-frequency terms and half corresponds to negative frequency terms; the output  $y[k]$  is a complex value however we are only interested in the magnitude of the frequency terms which is the absolute value of the complex number. When the input is real valued, the negative frequency terms are the conjugate of the positive frequency terms, therefore half of the output can be discarded and the computations will be done on the first half, multiplying the end result by a factor of 2. In our implementa-

tion, we used the Fast Fourier Transform implementation provided by scipy inside scipy.fft package.

Audacity software was used to inspect the spectrograms of the processed signals. We noticed that the maximum frequencies values reach around 5000Hz. The sample used when recording is 44100 Hz so we have decided to resample the signal to a lower sampling rate. According to the Nyquist theorem, to accurately represent a the highest frequency component of a signal ( no aliasing), the signal needs to be sampled at minimum twice the rate of that frequency. Considering this, in our case we have decided to resample our signal at the frequency of 10000Hz.

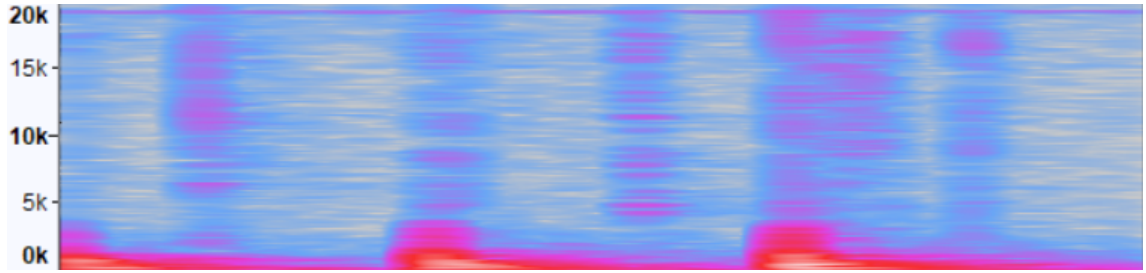


Figure 3.5: Audacity spectrogram view of the signal

W. Zeng and Mcloughlin [2013](#) used the energy ratio of certain frequency bands in relation to the total energy of the signal as a feature for the classification of watermelons.

### 3.4 Experiments and Results

The machine learning model was developed using python 3.8 with the keras api integrated in the Tensorflow 2 framework (version 2.5). ANN models were trained for each feature vector and evaluated separately. The architecture used to train the models had two hidden layers and one node in the output layer. In order to evaluate performance we used K-fold cross validation. Typically the dataset is divided into a training set and a testing set. The ANN uses the training set data to adjust its parameters in order to maximize a specified loss function; our problem is a binary classification problem so the loss function used was binary cross-entropy loss. After the training phase, the model's performance is then evaluated on the test set which contains input data that the model did not previously encounter in the training step. K-fold cross validation is a method where the dataset is split into K folds and K different models are constructed, each uses one of the K folds as a test set and the rest for training. This approach gives a better evaluation of the model's ability to generalize when applied on novel data and can be used to find better hyperparameters for general performance. The figure below is an illustration of this concept.

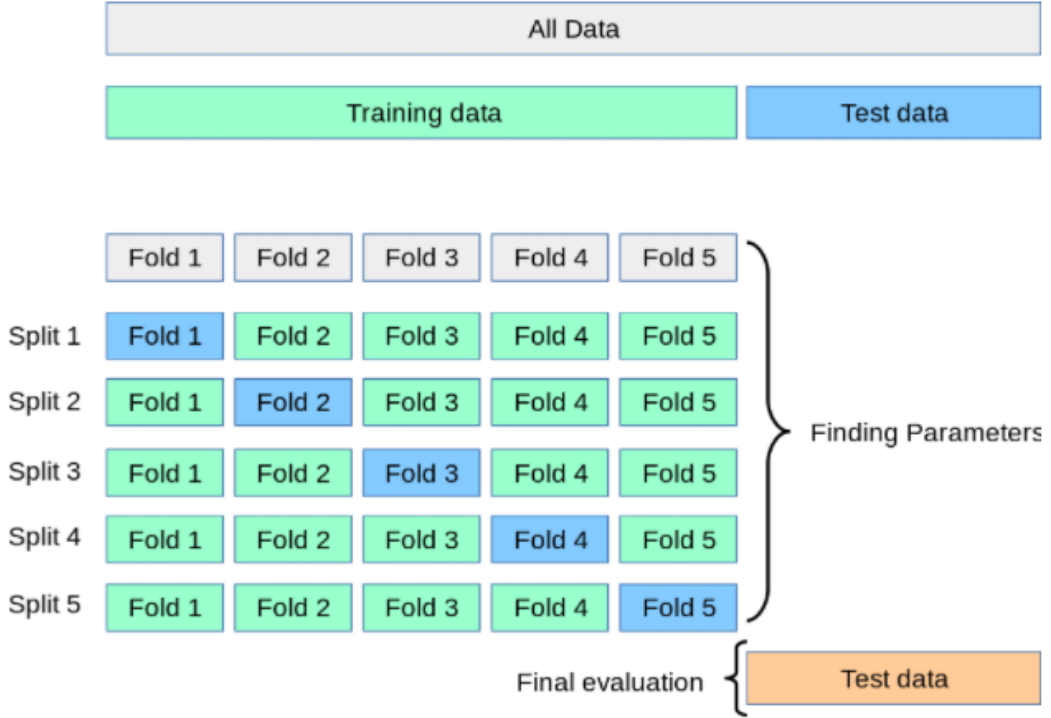


Figure 3.6: Sklearn cross validation illustration

Considering that our data-set is not particularly large 4 fold cross validation was used instead of the typical 5.

### 3.4.1 ANN training

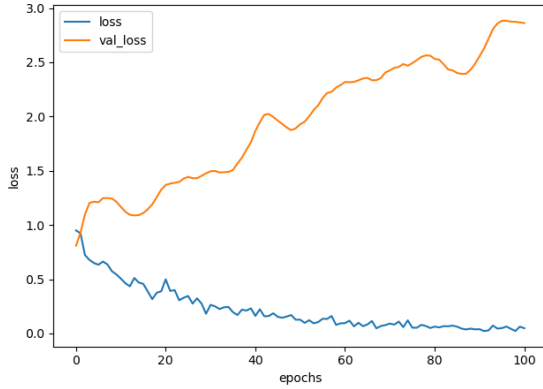
For each feature ANN models were developed and evaluated to assess the feature's potential as a discriminator for classifying watermelon ripeness. To evaluate performance the mean accuracy of the scores registered during cross validation was used. Using the STE feature vector, we obtained a mean accuracy of 0.56%. The ZCR feature vector had a corresponding mean accuracy of 0.49%. The power/energy feature vector was obtained by the following transformation on the FFT output:

$$f'[k] = 2 \cdot \frac{|y[k]|^2}{N}, k = 0, 1, \dots, N/2$$

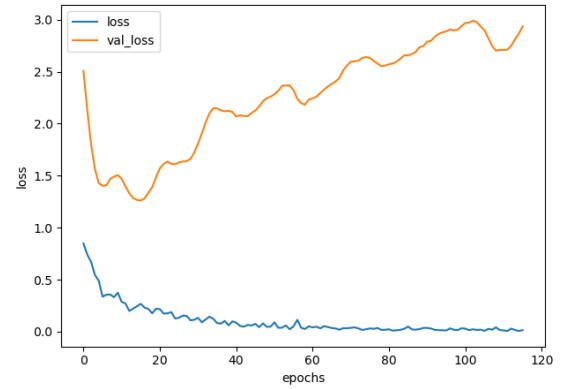
$$f[k] = \frac{f'[k]}{\sum_{i=0}^{N/2} f'[i]}, k = 0, 1, \dots, N/2$$

$y$  - FFT output vector  
 $N$  - length of signal  
 $f'$  - intermediary output vector  
 $f$  - output feature vector

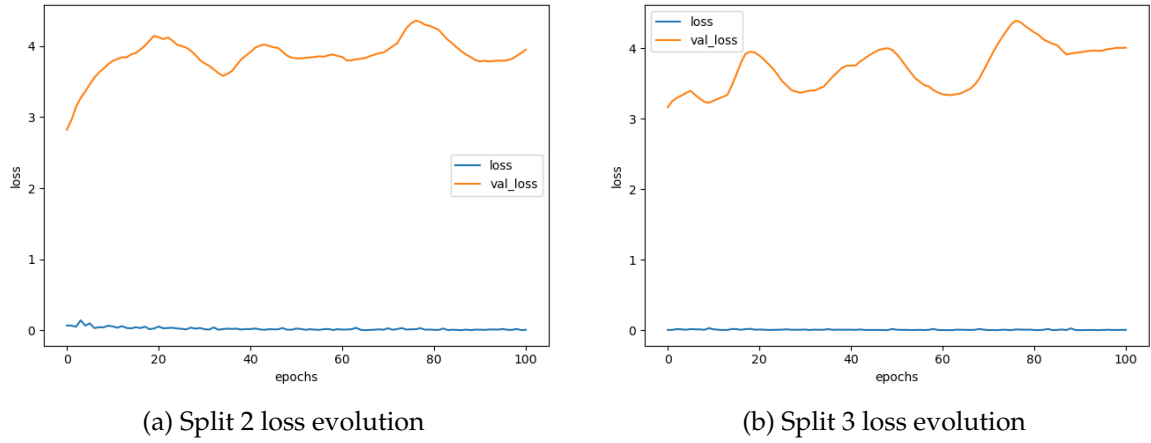
This resulted in around 1250 frequency bins ( from 0-5000Hz frequency), which means 1250 features for each sample. Using these features the ANN scored a mean accuracy of 70%, however it was observed that the model had a tendency to over-fit. Over-fitting is a problem that appears when a model adjusts its parameters to the training data to the extent that it negatively impacts performance on new data, leading to a lack of generalization of the learning process. Artificial neural network are prone to this problem because of the large amount of adjustable parameters; the problem is aggravated when the dataset used is rather small. The graphics below shows the loss evolution over epochs for each of the four splits during cross validation.



(a) Split 0 loss evolution



(b) Split 1 loss evolution



We are using binary cross-entropy loss which increases when the predicted labels diverge from the actual labels. The graphics show that the training loss decreases while the validation loss increases which means that the model is overfitting because it is not able to generalize to new data outside the training set. An attempt to balance this was to add dropout layers and increase the dropout rate however this change was not effective in eliminating overfitting. Another possible solution considered was to reduce the number of features to fewer, more relevant ones. This will mean the model has less adjustable parameters which will make overfitting more difficult while also potentially improving performance by eliminating irrelevant features.

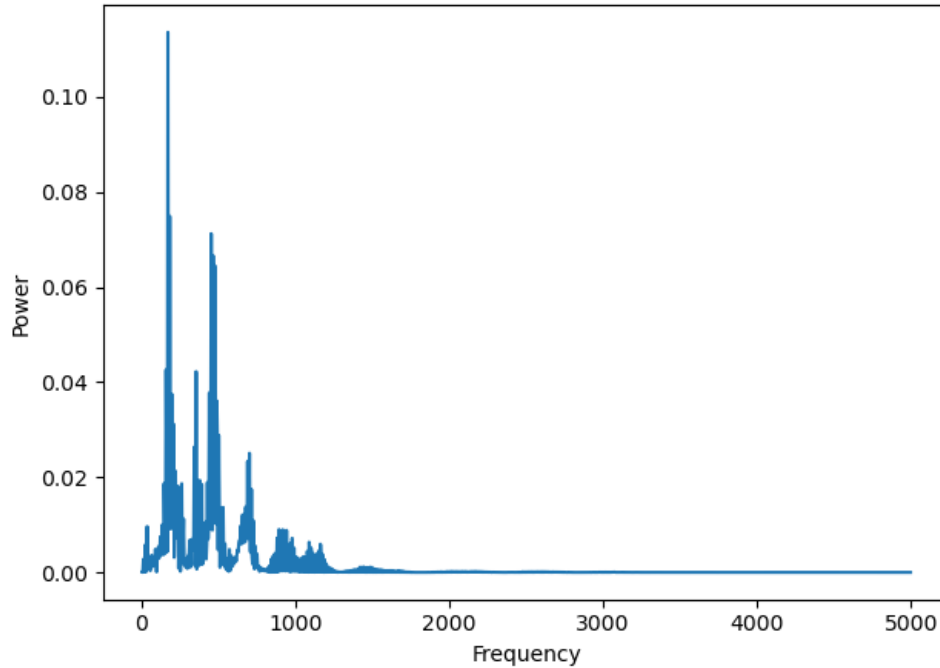
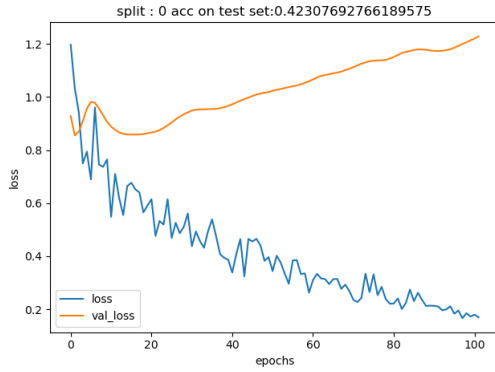


Figure 3.7: Power Frequency plot

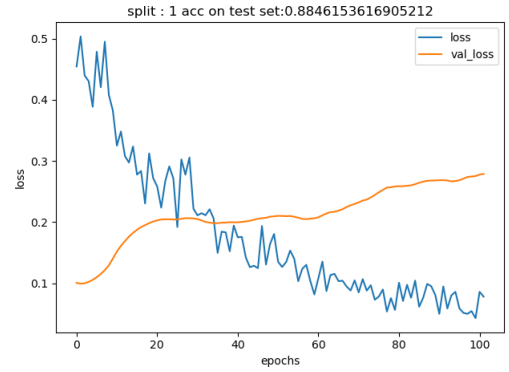
Examining the power spectrum we can see that most of the signal's energy is

concentrated in the lower frequency bands. Based on this observation a model was trained using the first 500 features, which correspond to the frequency range 0-2000Hz.

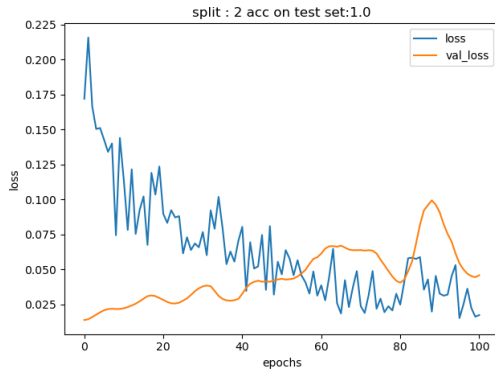
Using this feature vector resulting in a high mean accuracy of classification of 82%, however depending on the train test split results varied significantly, with the lowest performing ANN achieving 42% accuracy on the test set, while the highest one scored a 100% on the test set. The graphics below show the development of the ANN based on train/test split using the 500 features vector.



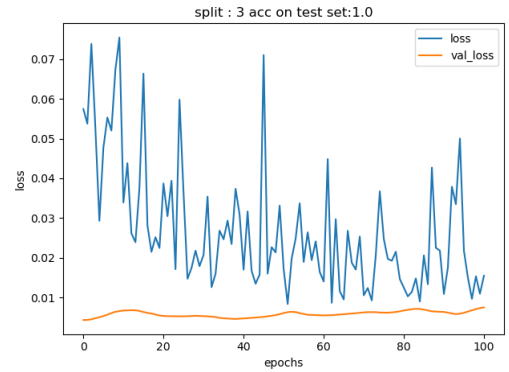
(a) Split 0 loss evolution



(b) Split 1 loss evolution



(a) Split 2 loss evolution



(b) Split 3 loss evolution

This much of a difference in performance based on train/test splits indicates that the trained model is still not able to generalize well when classifying new data. In order to further reduce the number of features and select only the most relevant ones we have used a genetic algorithm.

### 3.4.2 Genetic algorithm - Feature selection

A genetic algorithm is a search heuristic that is inspired by Charles Darwin's theory of natural evolution. We are interested in searching the frequency bands that minimize the mean loss of the ANN trained on the four train/test splits.

#### **Solution representation**

Each solution candidate (chromosome), determines which frequency bands will be chosen when training the ANN, which means the number of nodes in the input layer will differ between chromosomes. Internally this is represented as a binary array where each bit corresponds to a frequency term and frequencies corresponding to 0 valued bits will be ignored while frequencies corresponding to 1 valued bits will be used in the feature vector.

#### **Solution evaluation**

Each chromosome has a fitness associated with it which represents its performance. The evaluation of the fitness translates to training four neural networks, each using one of the train/test splits, where the input layer nodes are specified by the binary array as explained above. The ANN architecture is the same as specified in the previous section; two hidden layers each with a dropout layer. The model also uses early stopping which will stop training early if the validation error does not decrease for a set amount of epochs. The chromosome's fitness will be the average loss of these 4 models.

#### **Population initialization**

In the first epoch, each chromosome is initialised so that about 20% of the genes will take a value of 1. For each chromosome four neural networks will be trained each epoch, which makes running the genetic algorithm computationally intensive and takes a significant amount of time to run for a lot of epochs. Due to this fact, the population was limited to 101 chromosomes.

#### **Parents selection**

Each epoch, a new generation needs to be created. We used tournament selection: we randomly select  $k$  chromosomes of the population and the one with the best fitness is selected as the parent where  $k$  is specified as a hyperparameter which controls how much of the population is used in the tournament. For our experiments we used 30% of the population.



### Mutation & crossover operators

The mutation operator is used to diversify the chromosomes, search new solutions and re-introduce previously lost genetic material back in the population. The mutation operator used affects a gene by flipping the bit from a 0 to 1 or 1 to 0. This translates to incorporating a new frequency band in the feature vector or eliminating an existing one respectively. When mutation is applied, each gene has a probability  $pm$  of being affected by the mutation. For our experiments we used  $pm = 10\%$ .

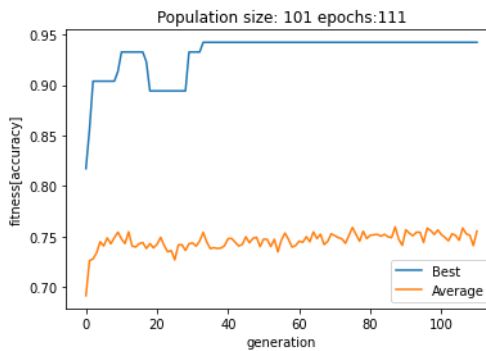
The crossover operator is applied to two chromosomes, which are called parents, in order to create two new ones, which are called children. We employed two point crossover in which two random indexes are chosen and the genes between them will be interchanged between the parents, creating two new chromosomes. Similarly to the mutation operator, when crossover is applied there is a probability  $pc$  that the parents are affected and return two new chromosomes or just a copy of the themselves. For our experiments we used  $pc = 90\%$ .

### Survivors selection

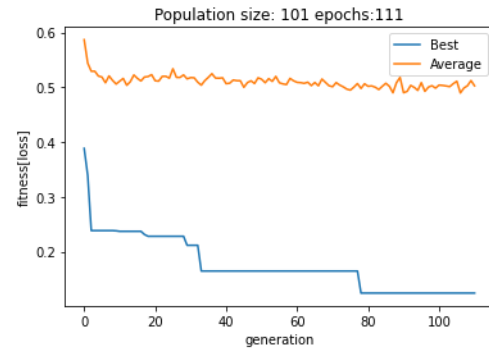
Each epoch, the chromosomes' fitness are evaluated and parents are selected until a new generation of children is constructed. An elitist variation was chosen for the selection of the survivors where each new generation replaces the old one entirely except the best five from the previous generation (in terms of fitness). This is done to ensure that the genetic material of the best solutions is not lost from one generation to another.

### 3.4.3 GA results

The GA algorithm ran for 111 epochs. The best chromosome obtained a mean accuracy of 94%. Below is a graphic showing the evolution of the GA population over the epochs.

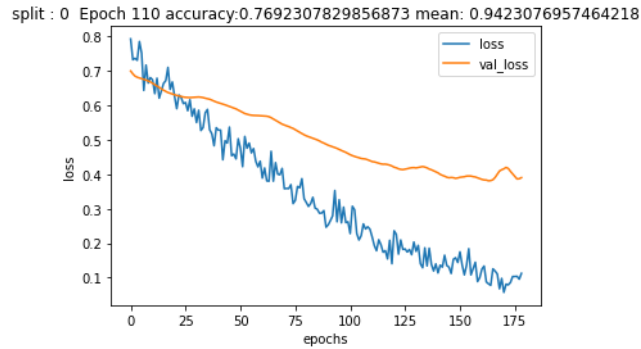


(a) Accuracy evolution

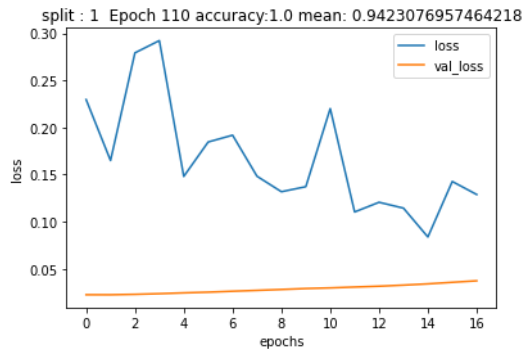


(b) Loss evolution

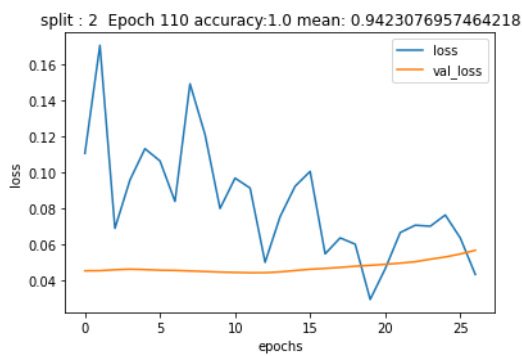
Examining the loss evolution graphics of the four ANNs it can be noted that the neural network reaches a high accuracy early on in the training and is stopped soon due to the early stopper. The ANN corresponding to the first split (split 0) starts off lower and reaches an accuracy of around 77% before training stops early.



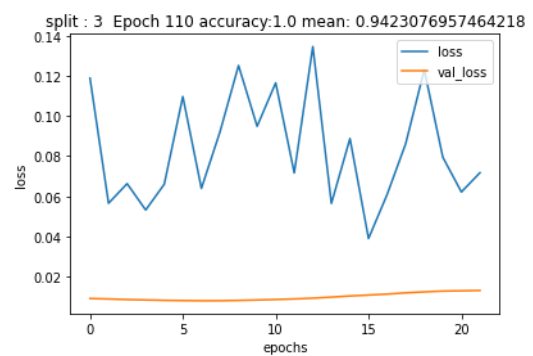
(a) Split 0 loss evolution



(b) Split 1 loss evolution



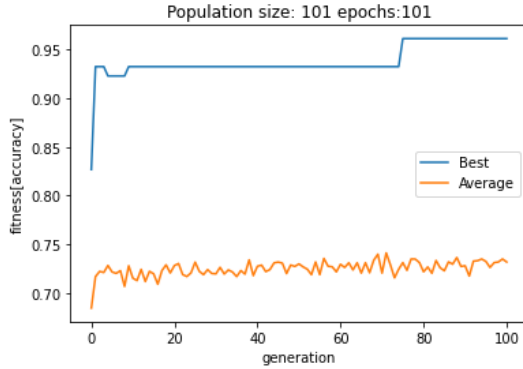
(a) Split 2 loss evolution



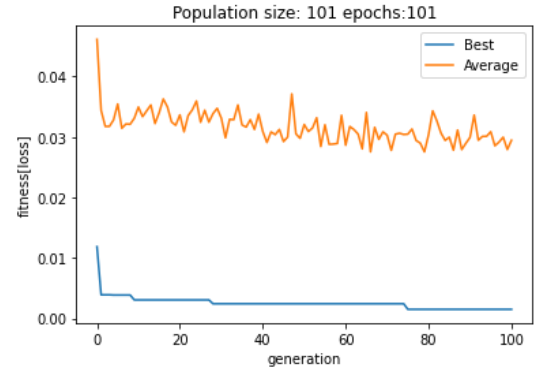
(b) Split 3 loss evolution

This gives a 17% difference in accuracy between the mean and the lowest one which is still quite significant. The effects of the train/test data split choice that can be observed indicate the possibility that the model is still over-fitting. This is challenging to confirm because the model does obtain a high accuracy on the test set and no other additional data is available for testing. Also the data set available is not particularly large; reducing the training set would potentially mean the training data will lack variety and will lead the model to overfitting.

To make the GA choose the frequencies that lead to a model able to better generalize we have decided to run the GA again with a different fitness function. Instead of the average loss of the neural networks the fitness will be defined as the mean loss divided by the relative standard deviation of the loss. This should encourage more robust networks based on the idea that a model able to generalize better will have less variation in performance depending on the train/test split chosen. Another GA was run for 101 epochs, this time using the new fitness function mentioned above. After finishing the 101 epochs, the best chromosome had a mean accuracy of 96%. Below are the graphics showing the GA evolution over generations.

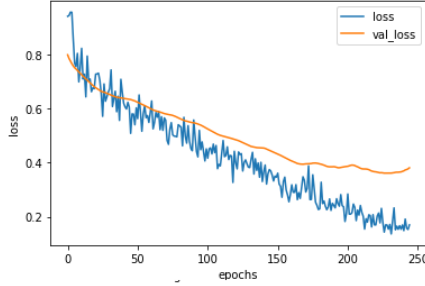


(a) Accuracy evolution



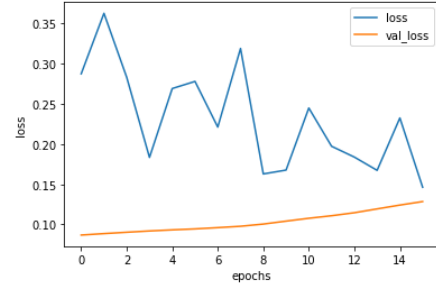
(b) New loss evolution

split : 0 Epoch 100 accuracy:0.8846153616905212 mean: 0.9615384489297867



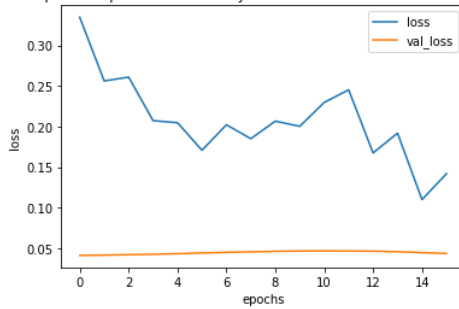
(a) Split 0 loss evolution

split : 1 Epoch 100 accuracy:0.9615384340286255 mean: 0.9615384489297867



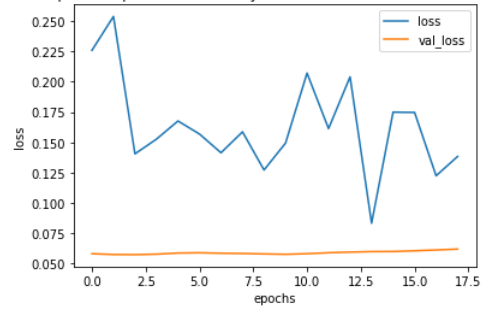
(b) Split 1 loss evolution

split : 2 Epoch 100 accuracy:1.0 mean: 0.9615384489297867



(a) Split 2 loss evolution

split : 3 Epoch 100 accuracy:1.0 mean: 0.9615384489297867



(b) Split 3 loss evolution

The training results are similar to the previous run; the training stops early as the model starts with a high accuracy, however this time the variation between the mean and the lowest performing ANN is significantly smaller at 8%, from 17% previously. In addition, the first split is stopped later than the one in the previous GA's best chromosome and reaches a higher accuracy. These indicate that the new fitness function was effective.

### 3.4.4 Final model

Due to poor accuracy the STE and ZCR features were discarded when developing the final model. The best chromosome of the GA model mentioned previously encodes the most relevant frequencies. Out of 500 frequencies, about 41% were selected. Below is a graphic where the frequencies selected are marked with a dot.

A neural network was trained using the features corresponding to the frequencies plotted above. Its performance was evaluated in multiple runs with random train/test splits. Performance varied depending on split, but was not able to consistently achieve the accuracy obtained by the best chromosome; when testing the neural network separately we found that drops to low 70% accuracy range were frequent. This poses the question about overfitting, once again. Given our current dataset it is difficult to assess this matter further. Our experiments show promising

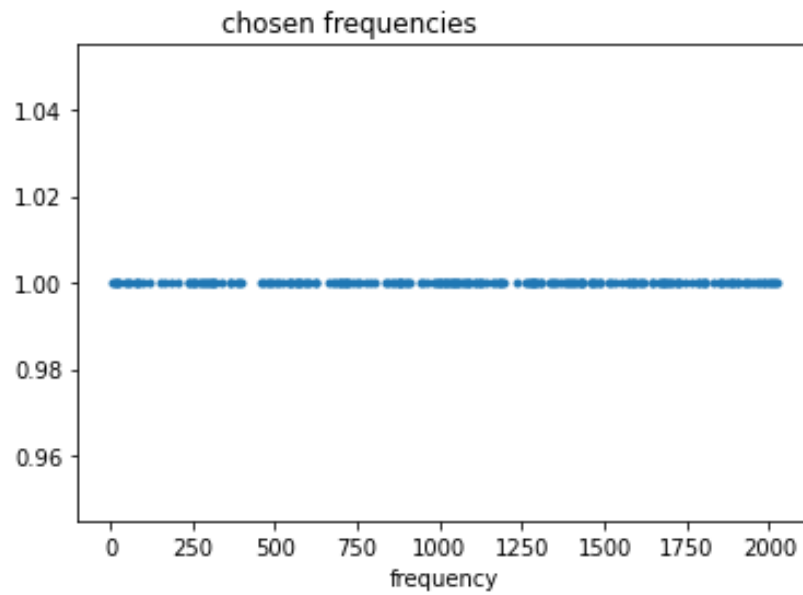


Figure 3.8: Selected frequencies

results for the proposed method of selecting relevant frequency bands, however a larger would be more suitable to be able to better verify the effectiveness of selecting discriminating frequency bands using genetic algorithms.

# Chapter 4

## Software Implementation

This chapter will describe the implementation of the proposed method in a mobile application as well as provide a user manual explaining the user interface.

### 4.1 Software design and architecture

We have chosen a client-server architecture for the application. The mobile clients will send raw recordings to a remote server API and will receive either a "ripe" or "unripe" response. This approach has a few advantages: because the mobile clients do not depend on the implementation of the ripeness evaluation model, improvements can be made easily without requiring updates for the application as long as the communication interface is maintained the same. Having all the computation be performed on the server is also notable taking into account the limited computation power of mobile devices. The disadvantage of this approach is that the mobile client will require an internet connection in order to connect to the server.

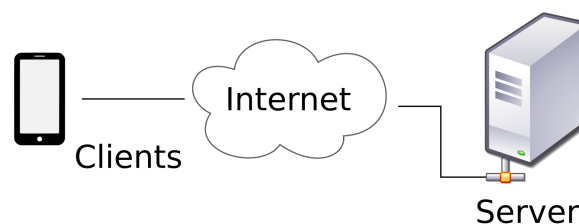


Figure 4.1: Client-Server model

### 4.1.1 Use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. The primary software requirements for our system are illustrated in the use case diagram below.

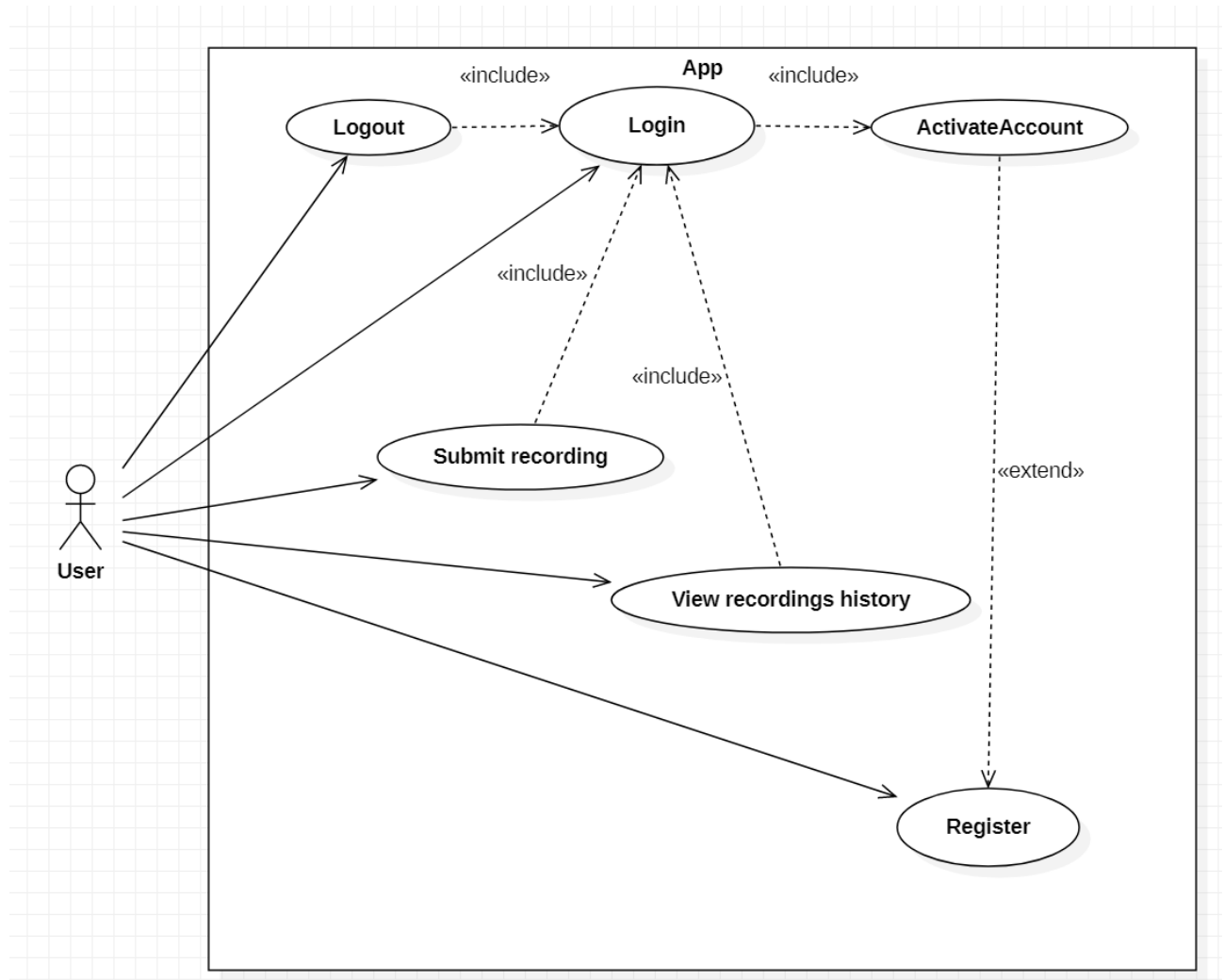


Figure 4.2: Use-case Diagram

### 4.1.2 Component diagram

The component diagram is a useful to get a visual representation of the breakdown of the structure of a system into distinct components, each responsible with high level functionalities.

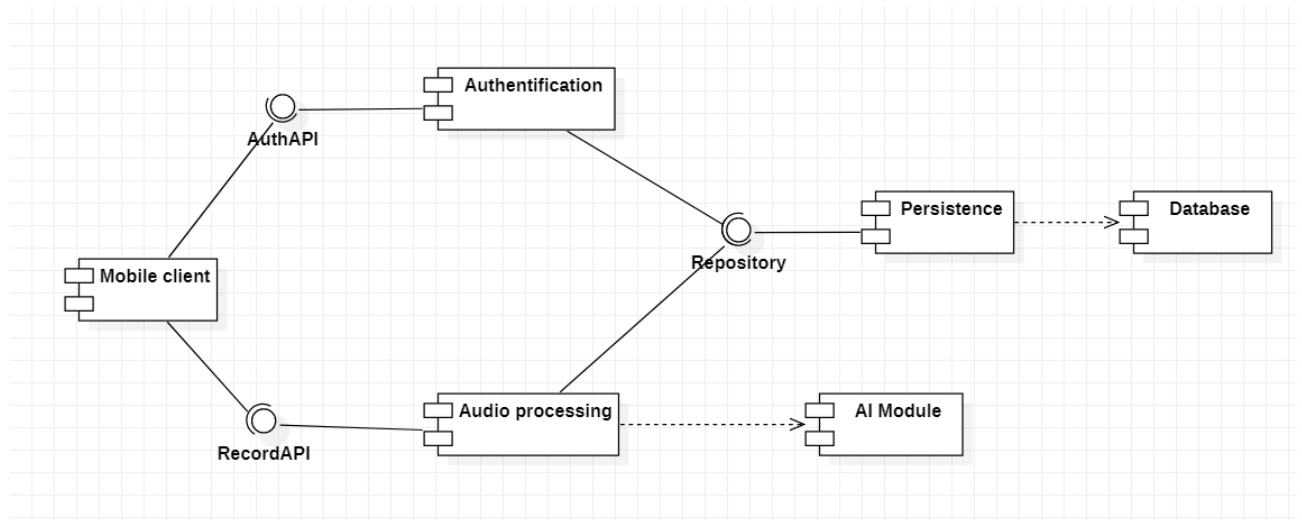


Figure 4.3: Components Diagram

### 4.1.3 Sequence diagram

A sequence diagrams illustrates the interactions between objects arranged in time sequence. It provides a visual representation of the data flow from the beginning of a use case scenario until the end. Below there are two sequence diagrams, each corresponding to the primary use case scenarios. One is for the "submit recording" scenario in which the user taps the watermelon, the app records it and sends the recording to the server which replies with a ripe/unripe response which is afterwards displayed to the user. The other one is for the "view recordings history" scenario in which the user presses the "recordings history" button; the app will request all the previously submitted recordings from the server and display them to the user.



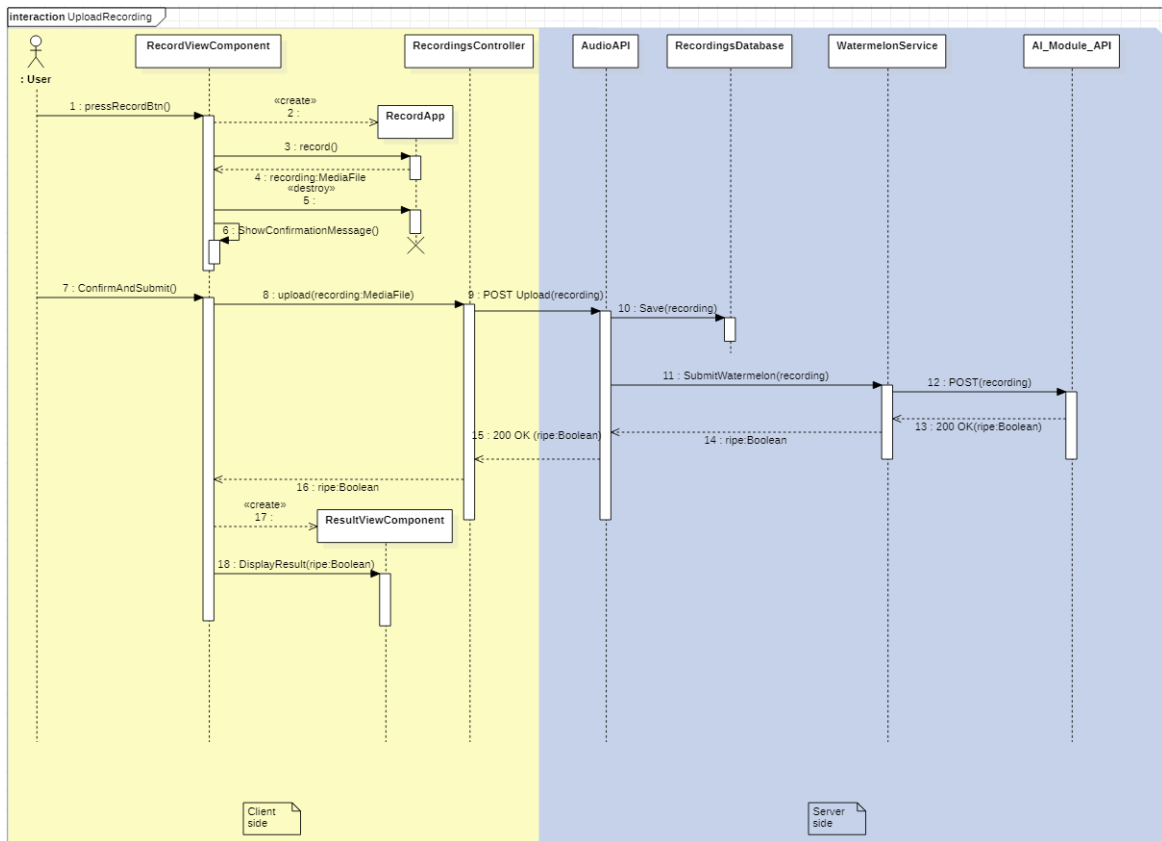


Figure 4.4: UploadRecording - Sequence Diagram

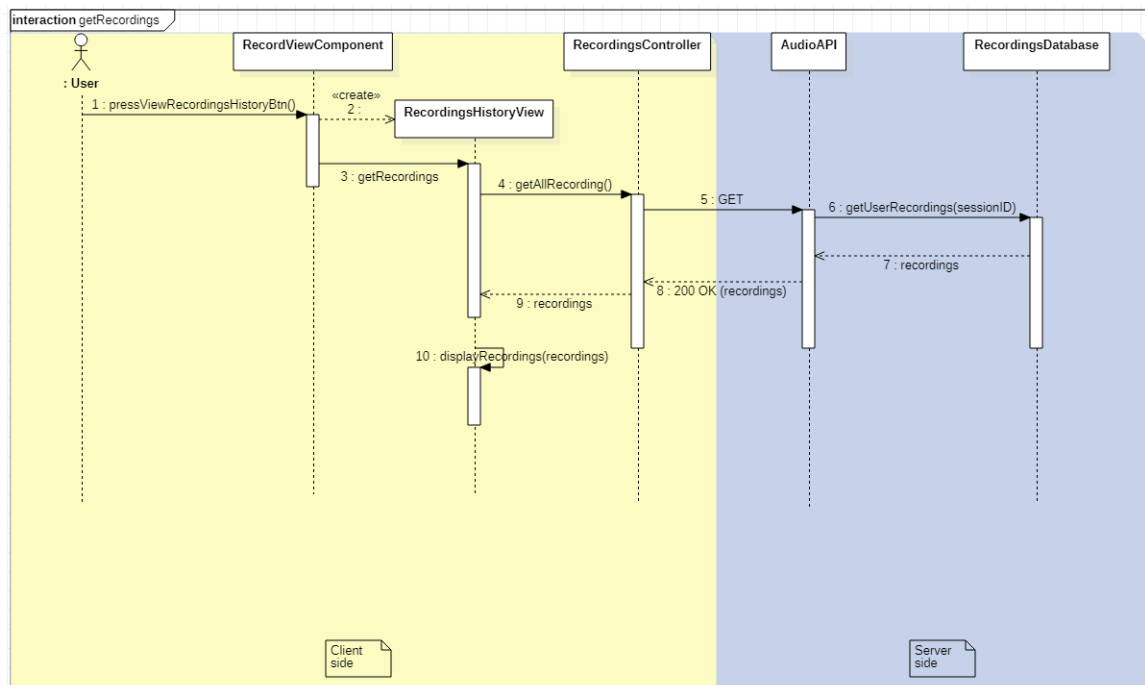


Figure 4.5: View recordings history - Sequence Diagram

## 4.2 Technologies used

This section briefly discusses about the languages and technologies used.

### 4.2.1 Ionic React

Ionic framework allows for cross platform mobile and web applications development using a single codebase. React is a javascript framework for building user interfaces that uses a component based architecture. It is one of the three frameworks supported by Ionic (also supports Vue and Angular frameworks) and is the chosen framework for our application.

### 4.2.2 Koa

Node js is a javascript runtime environment for back end applications. Koa is a web framework designed by the team behind the popular Express framework. Koa has the advantages of being lightweight and flexible allowing modern javascript syntax using async functions over callbacks. Our back end uses a Koa server to provide a Rest Api for the mobile client.

### 4.2.3 Database storage

Nedb is a fast and lightweight database for Node js which requires no other dependencies. It uses JSON format to store data and the api is a subset of MongoDB's. The main advantage it provides is the ease of use and that it doesn't require other dependencies.

### 4.2.4 Python

The AI module was written in Python version 3.8 using the keras api integrated in tensorflow 2.5. The fft implementation provided by the scipy.fft package was used. Data processing utilities from libraries such as scipy, numpy and librosa were also used. Matplotlib package was used for plotting graphs.

## 4.3 Testing & Specifications

### 4.3.1 API specifications

On the back-end side, the server provides a REST API for the mobile client. The table below describes the routes available for the client and their usage.

Route	Method	Description
/api/auth/login	POST	The client's credentials will be sent in the body of the request. If the login attempt is successful, the response contains a token for the client to use as identification for other requests
/api/auth/register	POST	The client's credentials will be sent in the body of the request. If the register operation is successful, an activation email will be sent to the client
/api/auth/activate/:id	PUT	After clicking the activation link in the email the mobile application will send a request to this route in order to activate the account
/api/recordings/	GET	This will request the previous recordings of the user. The user is identified based on the token received and the recordings will be sent in the body of the response.
/api/recordings/upload	POST	This route will handle submitted recordings of watermelons in order to check ripeness. The recordings are passed on to the AI module and a 'ripe' / 'unripe' response is sent back to the client.

### 4.3.2 Testing

The main purpose of testing activities is to verify whether a piece of software satisfies the application requirements. In this respect, unit tests and integration tests were written to test the node.js back end functionalities. Equivalence class partitioning was used as criteria for test coverage. A test case was developed for each equivalence class identified. The integration tests with the mobile API endpoints were carried out manually, checking the mobile application for the appropriate responses for each test case; the same was done for the AI module endpoint.

## 4.4 UserManual

This section provides documentation for the user interface with the aim to assist the user in navigating the interface without issues.

### 4.4.1 Login & Register

When the application starts the first screen is the login window. The user will enter his credentials and press login. If the user doesn't have an account he can register one by clicking the register text message below the login button. This will take the user to the register page where he will enter his email and password. An email containing an activation link will be sent for the user to the specified email. After the user clicks on the activation link the registration process will be complete and the user may proceed to login.

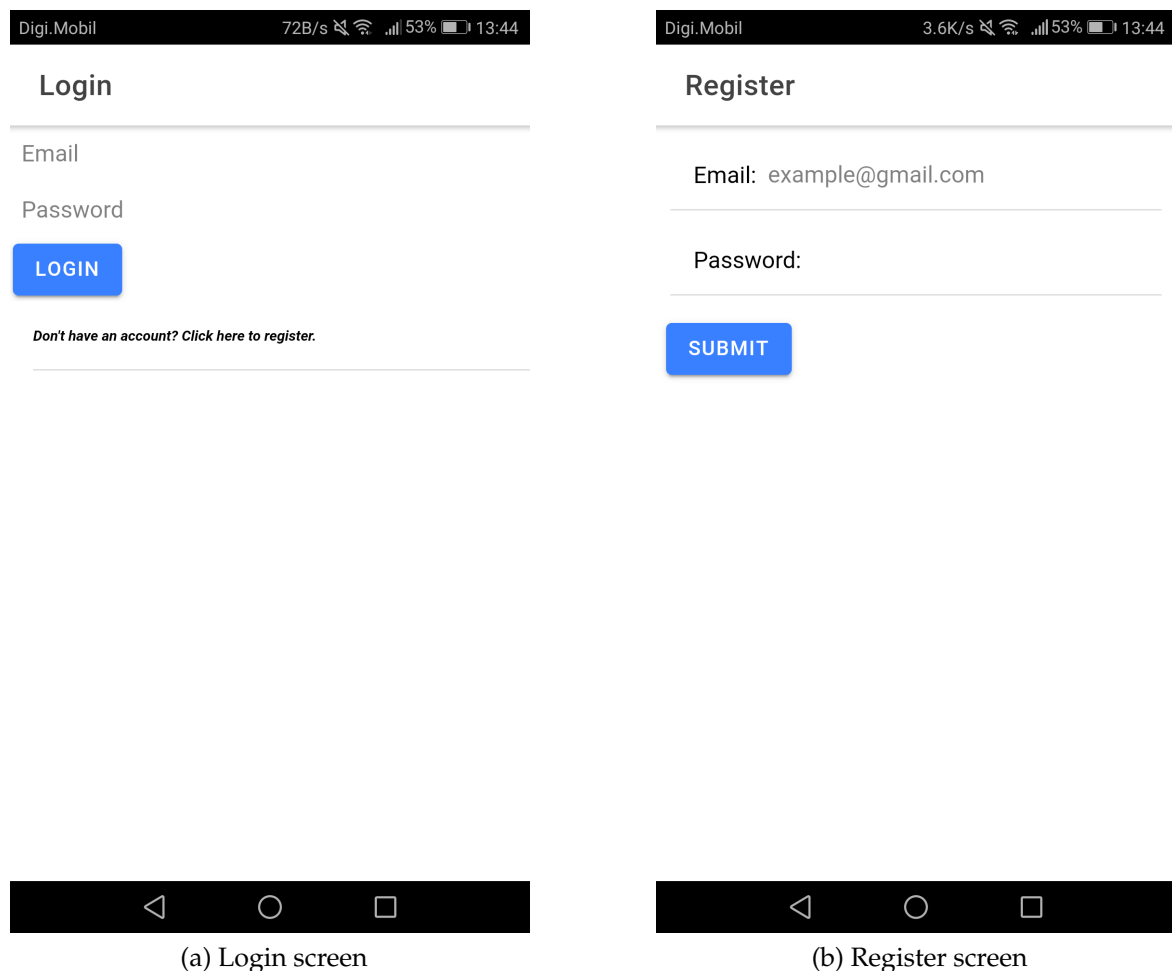


Figure 4.6: Login & Register screens

### 4.4.2 Home screen

Once logged in, the user will be redirected to the home screen page. Here the user can access the two main functionalities of the application: submit a new recording or view the previously submitted recordings.

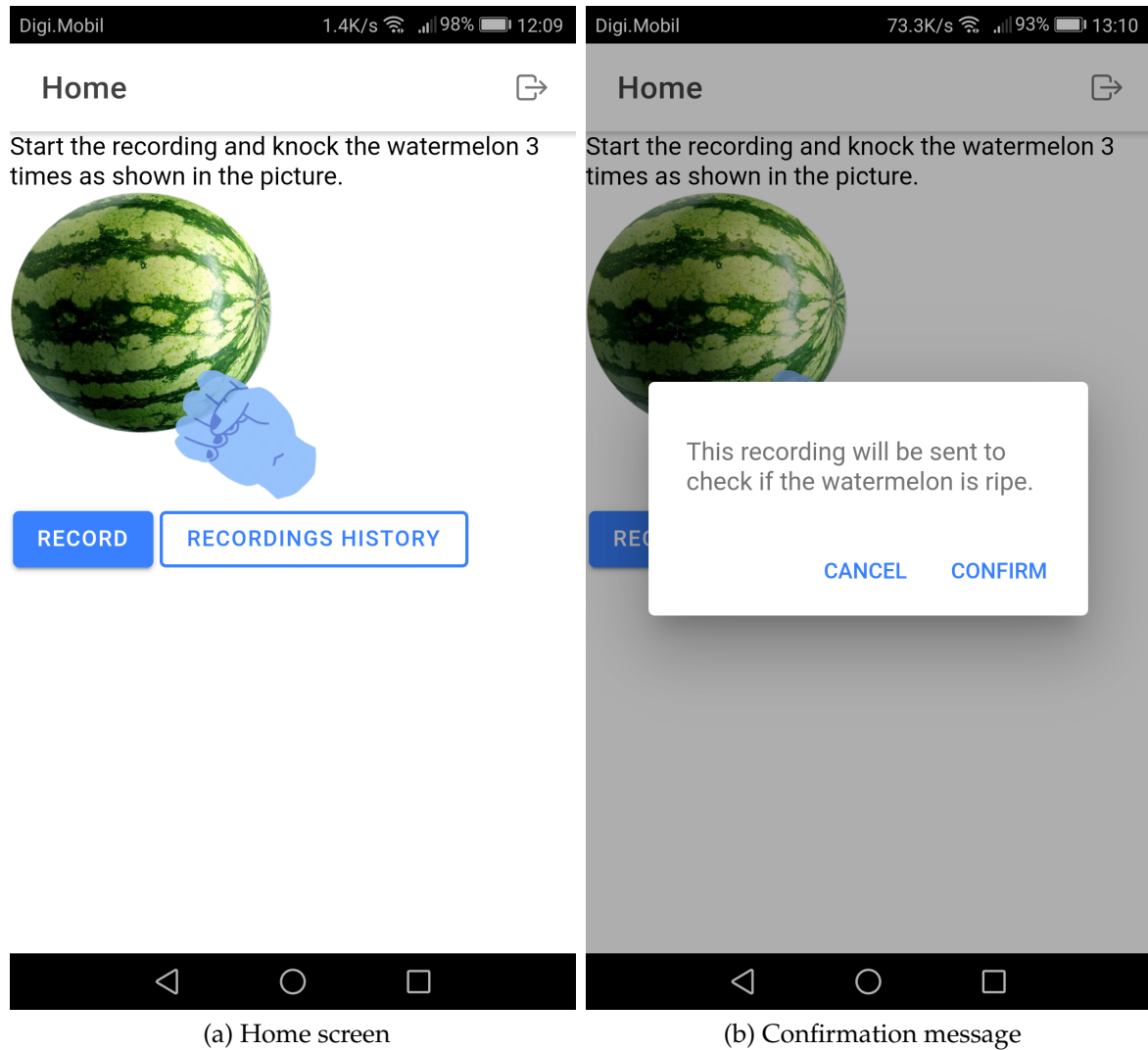


Figure 4.7: Home screen & confirmation

### 4.4.3 Submit Recording

To submit a recording the user will press the "Record" button. The user will knock the watermelon 3 times and end the recording. A confirmation message will appear. After confirmation the recording will be sent to the server. After a response is received from the server the app will display the result in a different screen.

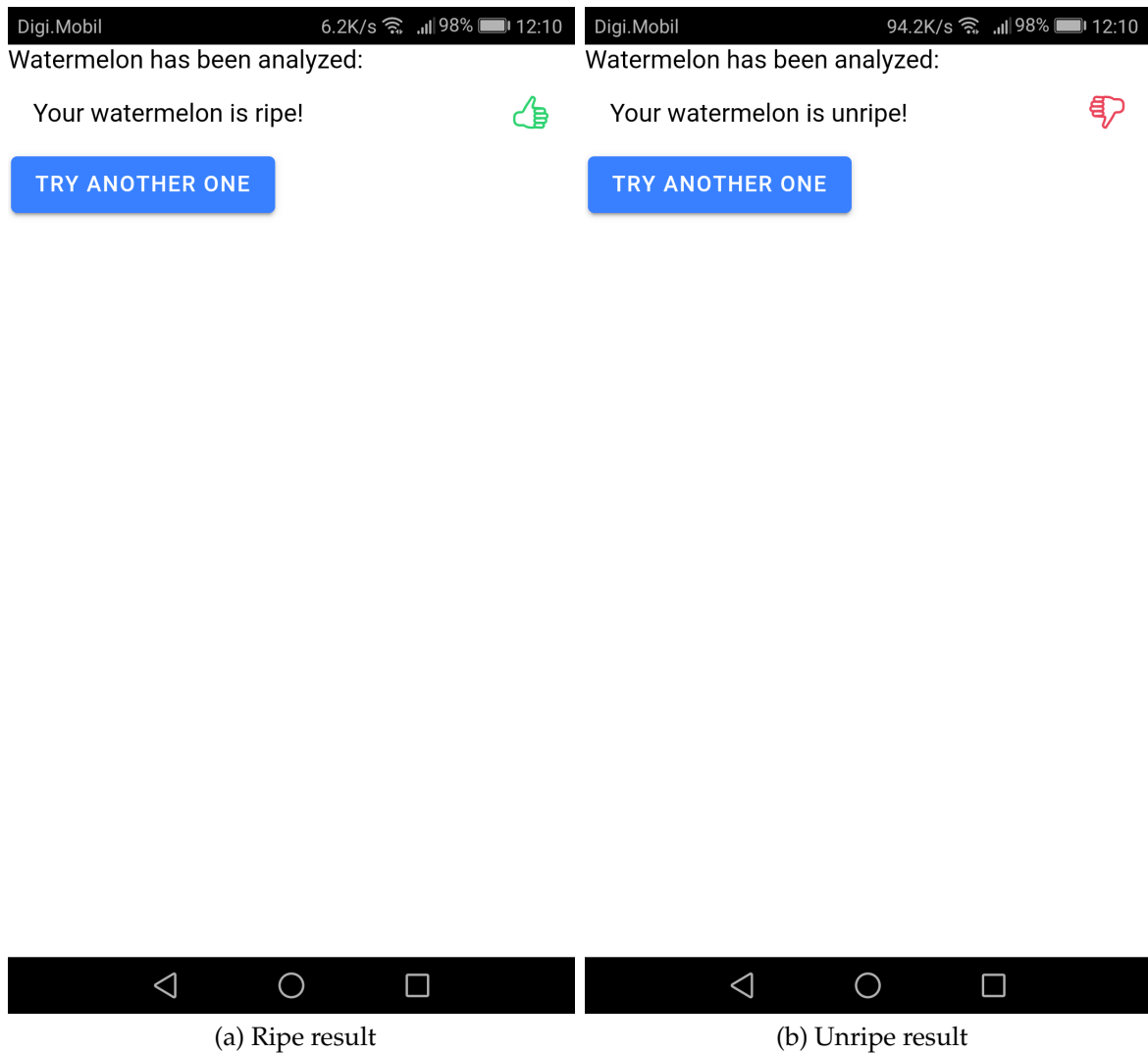
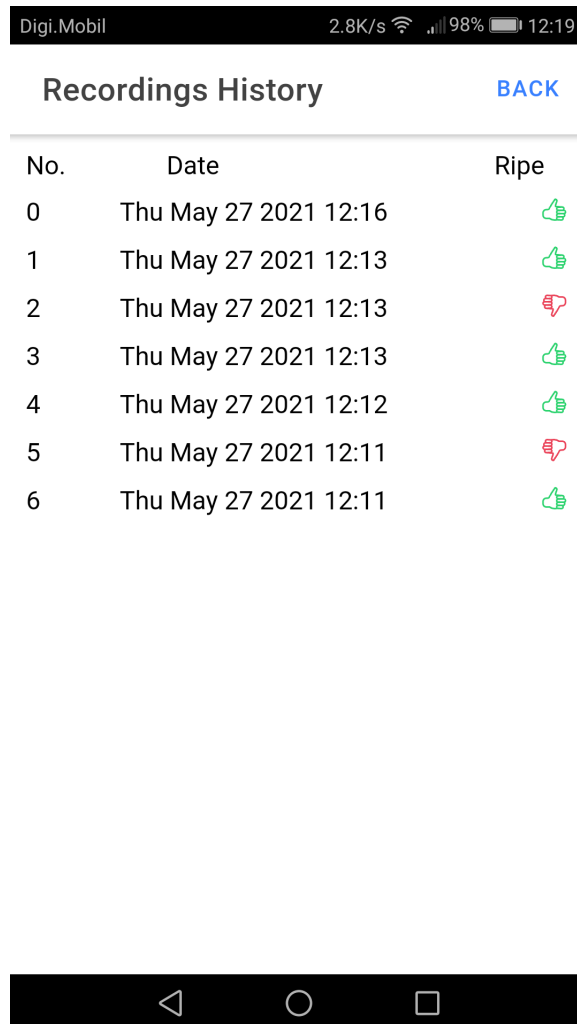


Figure 4.8: Result screen

#### 4.4.4 View recordings history

In the home screen view, the user can press the "Recordings History" button to be redirected to a new screen where he can view all the previously submitted recordings and their results.



No.	Date	Ripe
0	Thu May 27 2021 12:16	👍
1	Thu May 27 2021 12:13	👍
2	Thu May 27 2021 12:13	👎
3	Thu May 27 2021 12:13	👍
4	Thu May 27 2021 12:12	👍
5	Thu May 27 2021 12:11	👎
6	Thu May 27 2021 12:11	👍

Figure 4.9: Recordings history screen

# Chapter 5

## Conclusions

The current paper's objective was to investigate watermelon ripeness classification methods based on acoustic sounds recorded using mobile devices using machine learning techniques. The performance of artificial neural networks trained on acoustic features derived from the frequency domain was assessed and a genetic algorithm was proposed in order to select the most relevant frequency bands.

Results showed that neural networks trained using the frequency bands chosen by the genetic algorithm were able to achieve higher accuracy than neural networks trained on the entire feature set. This was especially true for the GA variant where the fitness function also takes into account the relative standard deviation of the losses, which improved the robustness of the best Chromosomes' solution.

### 5.1 Personal contributions

A mobile application was developed to record and predict watermelon ripeness based on a client-server model. The application was built using Ionic framework which means it can be adopted to work cross-platform for both Android and IOS with minimal changes in code and configuration. A genetic algorithm was designed and experimented in order to select the most relevant features for classification. This method can be applied to improve performance and robustness of neural networks classification models. Another contribution is the gathering of recordings of real data from 104 watermelons in the summer of 2020 which were used in our experiments.

### 5.2 Further work

The data set is of utmost importance when training a machine learning model. Having a large and diversified data set increases the robustness and performance of



trained models. The data set used had 104 samples which is rather low when training artificial neural networks. A larger data set would help avoid overfitting and better assess the robustness of models and the proposed GA method.

Also, the method used to label the watermelons was to classify each as either ripe/unripe based on visual inspection and taste test. This, however is a subjective method and is prone to human error, introducing noise in the dataset. A better method would be an objective evaluation such as measuring the sugar content in the watermelon which would transform the problem from a classification problem to a regression problem.

The recordings used in our experiments were taken in a quiet setting, however the background noise might in a supermarket could have a negative effect on the model's ability to classify watermelons from a recording. The precise effect on accuracy could be tested, since the frequency components we are interested in mostly lay in the lower spectrum, a high-pass filter could potentially work to counter this problem.

Finally, the focus of our experiments were not to find the best parameters for the artificial neural networks but to investigate a general method of selecting relevant features. The best parameters combinations and layer architecture could be investigated comprehensively to optimise the networks.

# Bibliography

- Abebe, A. T. (2006). "Total Sugar and Maturity Evaluation of Intact Watermelon Using near Infrared Spectroscopy". In: *Journal of Near Infrared Spectroscopy* 14.1, pp. 67–70.
- B. Diezema, M. Ruiz-Altisent and B. Orihuel (2003). "Acoustic impulse response for detecting hollow heart in seedless watermelon". In: *Acta Horticulturae* 599.599, pp. 249–256.
- Baki, S. R. M. Shah et al. (2010). "Non-Destructive Classification of Watermelon Ripeness using Mel-Frequency Cepstrum Coefficients and Multilayer Perceptrons". In: *International Joint Conference on Neural Networks*, pp. 1–6.
- C. V. Greensill P. J. Wolfs, C. H. Spiegelman and K. B. Walsh (2001). "Calibration Transfer between PDA-Based NIR Spectrometers in the NIR Assessment of Melon Soluble Solids Content". In: *Applied Spectroscopy* 55.5, pp. 647–653.
- D. Jie L. Xie, X. Rao and Y. Ying (2014). "Using visible and near infrared diffuse transmittance technique to predict soluble solids content of watermelon in an on-line detection system". In: *Postharvest Biology and Technology* 90, pp. 1–6.
- Dull, G. G., G. S. Birth, et al. (1989). "Near Infrared Analysis of Soluble Solids in Intact Cantaloupe". In: *Journal of Food Science* 54.2.
- (1990). "Near-infrared spectrophotometry for measurement of soluble solids in intact honeydew melons". In: *HortScience* 25, p. 1132.
- Dull, G. G., R. G. Leffler, et al. (1992). "Instrument for nondestructive measurement of soluble solids in honeydew melons". In: *Transactions of the ASAE* 35.2, pp. 735–737.
- Flores, K. et al. (2008). "Prediction of Total Soluble Solid Content in Intact and Cut Melons and Watermelons Using near Infrared Spectroscopy". In: *Journal of Near Infrared Spectroscopy* 16.2, pp. 91–98.
- Ito, H., S. Morimoto, et al. (2002). "Potential of near infrared spectroscopy for non-destructive estimation of soluble solids in watermelons". In: vol. 588. 566, pp. 353–356.
- Ito, H. and J. Sugiyama (2002). "Nondestructive harvest time decision of melons by a portable firmness tester". In: *Acta Horticulturae* 579, pp. 367–371.

- J. Guthrie, B. Wedding and K. Walsh (1998). "Robustness of NIR Calibrations for Soluble Solids in Intact Melon and Pineapple". In: *Journal of Near Infrared Spectroscopy* 6.1, pp. 259–265.
- J. Mao Y. Yu, X. Rao and J. Wang (2016). "Firmness prediction and modeling by optimizing acoustic device for watermelons". In: *Journal of Food Engineering* 168.2, pp. 1–6.
- J. Sugiyama K. Otobe, S. Hayashi and S. Usui (1994). "Firmness Measurement of Muskmelons by Acoustic Impulse Transmission". In: *Transactions of the ASAE* 37.4, pp. 1235–1241.
- Kato, K. and M. Matuda (2004). "Prediction of Melon Fruits Softening Using Density and Transmitted Light". In: *Key Engineering Materials*, pp. 270–273, 1064–1060.
- Koc, A. B. (2007). "Determination of watermelon volume using ellipsoid approximation and image processing". In: *Postharvest Biology and Technology* 45.3, pp. 366–371.
- Lee, K. J. et al. (2006). "Internal Quality Estimation of Watermelon by Multiple Acoustic Signal Sensing". In: *Key Engineering Materials* 321–323.1209–1212.
- Marita, C. (1996). "Case Study: Quality Assurance for Melons". In: *Perishables Handling Newsletter Issue No. 85*, pp. 10–12.
- N. A. Syazwan, S. R. M. Shah Baki and M. T. Nooritawati (2012). "Categorization Of Watermelon Maturity Level Based On Rind Features". In: *Procedia Engineering* 41, pp. 1398–1404.
- N. Jamal Y. Ying, J. Wang and X. Rao (2005). "Finite element models of watermelon and their applications". In: *Transactions of the CSAE* 21.1, pp. 17–22.
- N. Ozer, B. A. Engel and J. E. Simon (1998). "A multiple impact approach for non-destructive measurement of fruit firmness and maturity". In: *Transactions of the ASAE* 41.3, pp. 871–876.
- P. R. Armstrong, M. L. Stone and G. H. Brusewitz (1997). "Nondestructive acoustic and compression measurements of watermelon for internal damage detection". In: *Applied Engineering in Agriculture* 13.5, pp. 641–645.
- R. Abbaszadeh A. Moosavian, A. Rajabipour and G. Najafi (2015). "An intelligent procedure for watermelon ripeness detection based on vibration signals". In: *Journal of Food Science and Technology* 52.2, pp. 1075–1081.
- S. O. Nelson W. Guo, S. Trabelsi and S. J Kays (2007). "Dielectric spectroscopy of watermelons for quality sensing". In: *Measurement Science and Technology* 18, pp. 1887–1892.
- S. R. M. Shah Baki A. R. Farah Yasmin, M. Y. Ahmad Ihsan and K. Shazana (2009). "Non-destructive Watermelon Ripeness Determination Using Image Processing and Artificial Neural Network (ANN)". In: *International Journal of Computer and Information Engineering* 3.2, pp. 391–395.

- Saito, K. et al. (1996). "Application of magnetic resonance imaging to non-destructive void detection in watermelon". In: *Cryogenics* 36.12, pp. 1027–1031.
- Sugiyama, J. et al. (1998). "Melon Ripeness monitoring by a portable firmness tester". In: *Transactions of the ASAE* 41.1, pp. 121–127.
- Tian, H., Y. Ying, H. Lu, et al. (2007). "Study on predicting firmness of watermelon by Vis/NIR diffuse transmittance technique". In: *Spectroscopy and Spectral Analysis* 27.6, pp. 1113–1117.
- Tian, H., Y. Ying, H. Xu, et al. (2009). "Study on Vis/NIR Spectra Detecting System for Watermelons and Quality Predicting in Motion". In: *Spectroscopy and spectral analysis* 29.6, pp. 1536–1540.
- W. Zeng X. Huang, S. M. Arisona and I. Mcloughlin (2013). "Classifying watermelon ripeness by analysing acoustic signals using mobile devices". In: *Personal and Ubiquitous Computing* 18, pp. 1753–1762.
- X. Chen, P. Yuan and X. Deng (2018). "Watermelon ripeness detection by wavelet multiresolution decomposition of acoustic impulse response signals". In: *Postharvest Biology and Technology* 142, pp. 135–141.
- Y. Wei, X. Rao and B. Qi (2012). "Acoustic detecting system for sugar content of watermelon". In: *Transactions of the Chinese Society of Agricultural Engineering* 28.3, pp. 283–287.