Apr 26th, 3:15 PM - 2:30 PM

Watermelon Sweetness Prediction Based on Sound Waves

Kyla Williams
Ouachita Baptist University

Rafael Bustillo
Ouachita Baptist University

Watermelon Sweetness Prediction Based on Emitted Vibrational Frequencies

Rafael Bustillo, Kyla Williams, & Dr. Kevin Cornelius

Ouachita Baptist University, Arkadelphia, AR 71998

Introduction

The current project was focused on gathering audio signals from thumping watermelon to improve the our correlation between the sound produced and the sugar concentration of the watermelon. The Audio signals are converted to harmonic frequencies using the Fast Fourier Transform (FFT) and then analyzed, along with a few physical properties of the watermelon, to find the coefficients of a linear regression to predict the measured Brix value of a watermelon. This process is currently being performed in MATLAB on a PC, but work is being done to incorporate it into an iPhone app.

Data Collection

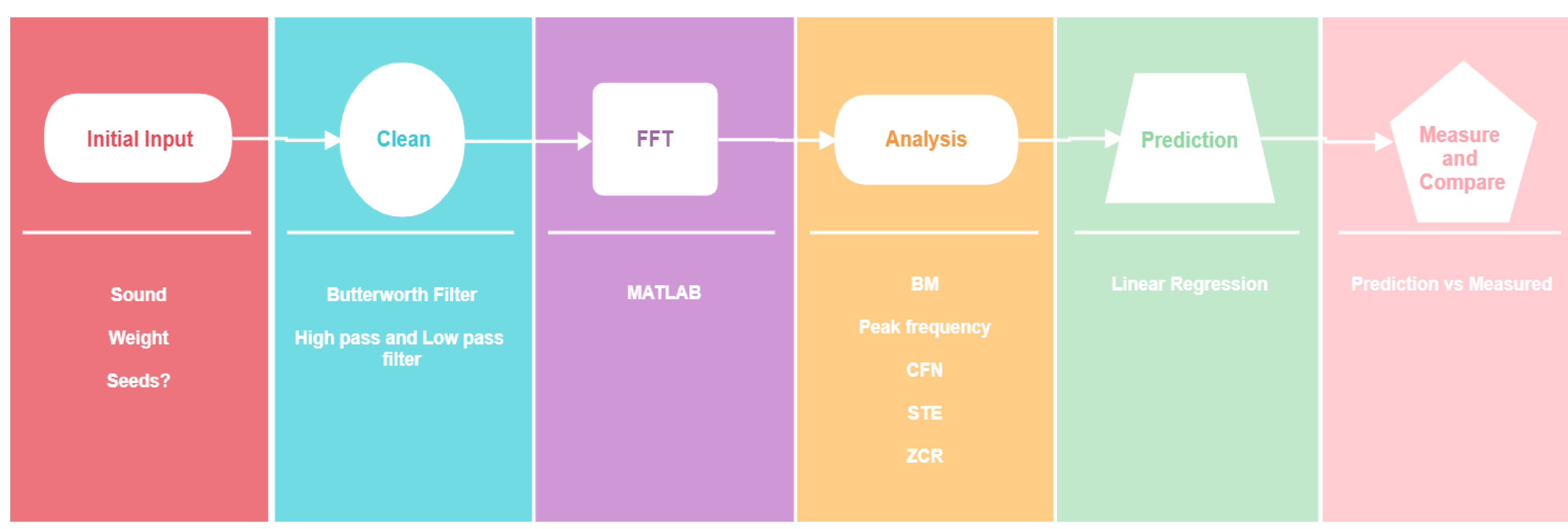

Figure 1. Flow Chart of the data collection and analysis process

- Record external characteristics
- Use the thumping apparatus to thump the watermelon
- Record the thump and determine the FFT in MATLAB
- Record various calculated parameters into Excel
- Cut open the watermelon and record internal characteristics
- Take a core from the melon for juicing
- Determine sweetness using Brix Refractometer

Figure 2. Thumping apparatus

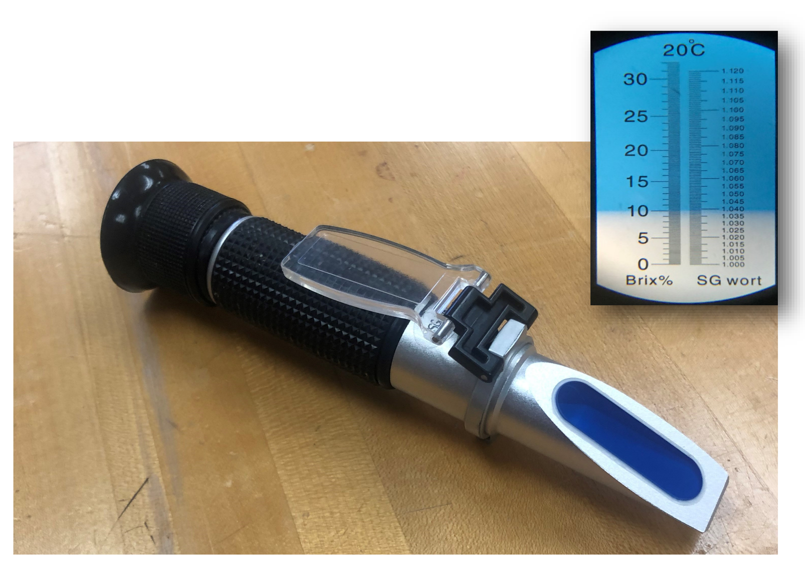

Figure 3. Brix Refractometer used to determine sweetness

Analysis

Data was collected from 70+ watermelons. MATLAB creates an FFT graph of the thump, as seen in figures 4 and 5. Various characteristics of this graph (*such as a certain band of frequencies*), along with weight and seed content, were used to construct a linear regression model in Excel that could predict the Brix value of the watermelon before cutting it open. Once an initial model with coefficients was determined, MATLAB code was written to include a predictor as part of the analysis. The prediction was added to the FFT graph to give us an idea of what should be measured.

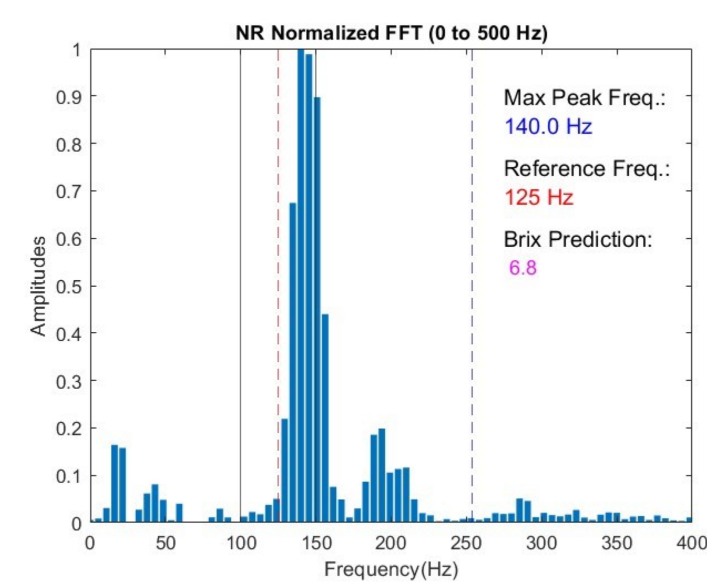

Figure 4. FFT of a watermelon that was a 6.5 on the brix scale. The prediction was a 6.8.

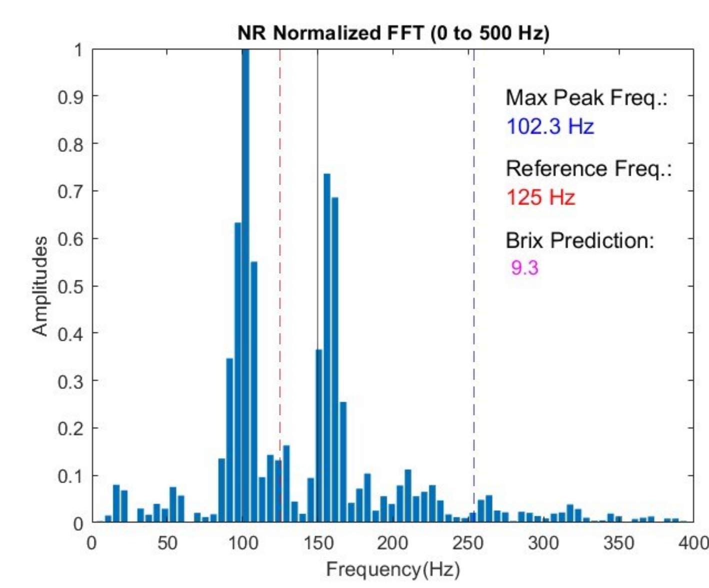

Figure 5. FFT of a watermelon that was a 8.8 on the brix scale. The prediction was a 9.3.

More data yielded better coefficients, which resulted in closer predictions. Our goal is to predict within 0.5 of the actual brix value (~5% error). The above examples fit that goal. Over the summer, 80% of our predictions had errors less than 12%, with an overall average of 7%.

Road Brix	Lab Brix	Real Brix
7.9	7.2	8
7.8	5.5	7
7.2	7.3	7
8.8	8.8	10
7.9	7.7	7.4

Figure 6. Table comparison of prediction values

The MATLAB code allowed us to field test watermelons in different noisy environments and predict Brix values before testing them in the lab under more controlled conditions. The table to the left shows a comparison of some of our field testing results.

App Development

- Designing user interface
- Recording audio signal
- Cleaning audio signal
  - Filters
- Processing audio signal
  - FFT and others
- Prediction

Figure 7. Snippet of app development in XCode

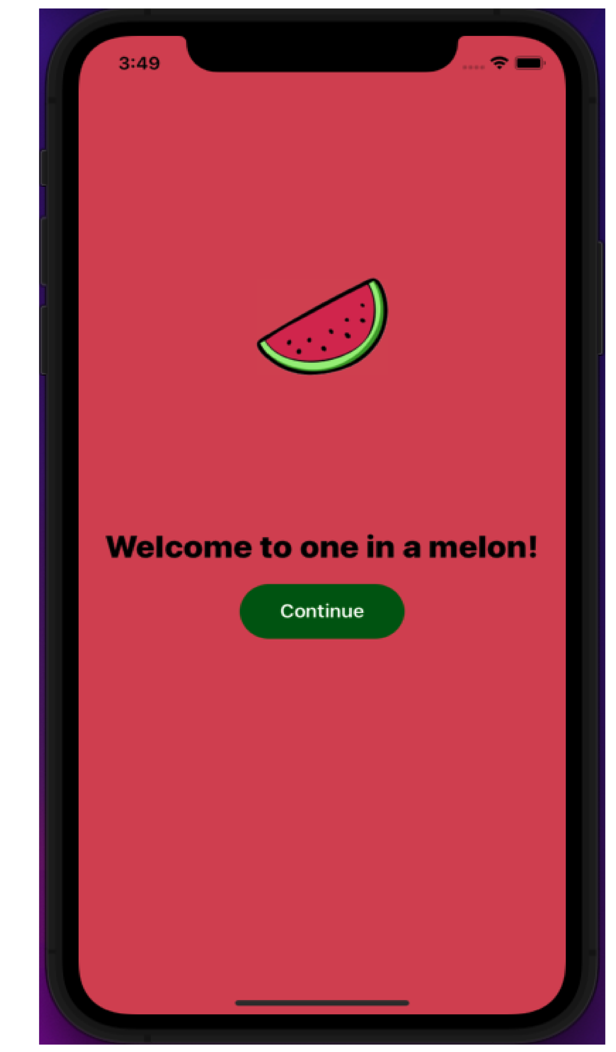

Figure 8. Screenshot of iPhone app welcome page.

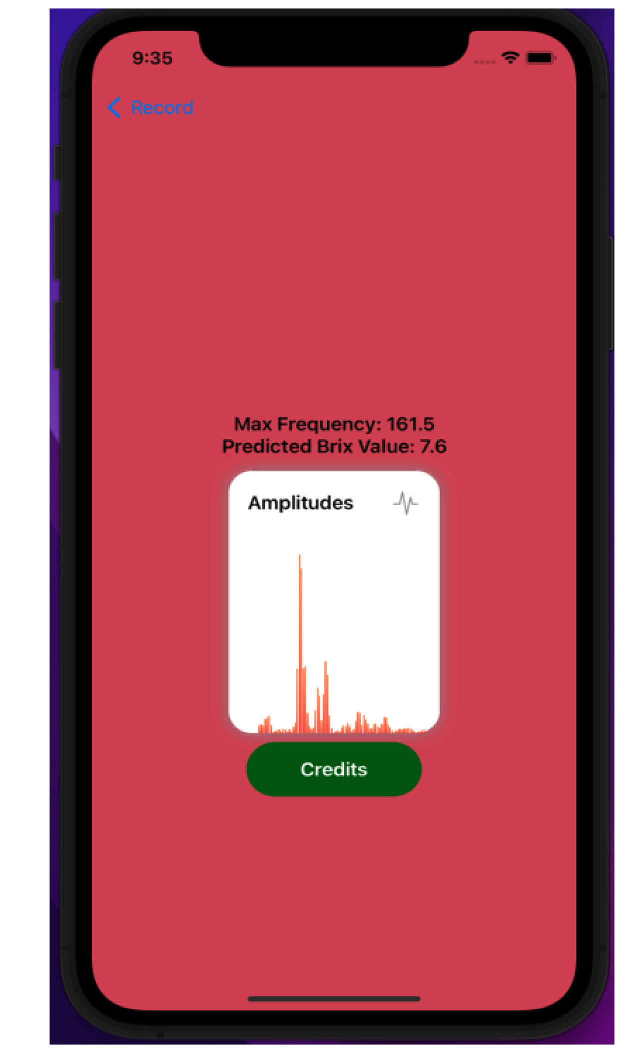

Figure 9. Screenshot of iPhone app results page.

Progress this summer resulted in almost finishing the coding for the iPhone app. Currently, the app consists of several screens, including an instructions screen, recording screen and results screen. Figure 8 displays the welcome screen of the app. Figure 9 shows the results screen that is displaying the max frequency (Hz) of the FFT, the predicted Brix value, and a graph of the FFT.

Conclusion and Future Work

The current predictive model is working very well and producing errors less than 12%. More tests need to be done to improve our regression coefficients and predictive results. In addition, testing needs to be done on watermelons that are underripe or overripe to help the predicter be more accurate for a wider range of melon conditions.

While developing the app, we found out that MATLAB and the iPhone have different algorithms to filter the audio signal and calculate the FFT. This resulted in some variance between the predictions on the iPhone and in MATLAB. Additional work is needed in an attempt to close the gap between these two codes. Once the codes are in alignment, time can be spent in polish the app's user interface.

Acknowledgements

. Dr. Cornelius
. OBU Physics Department
. OBU Patterson Summer Research Grant