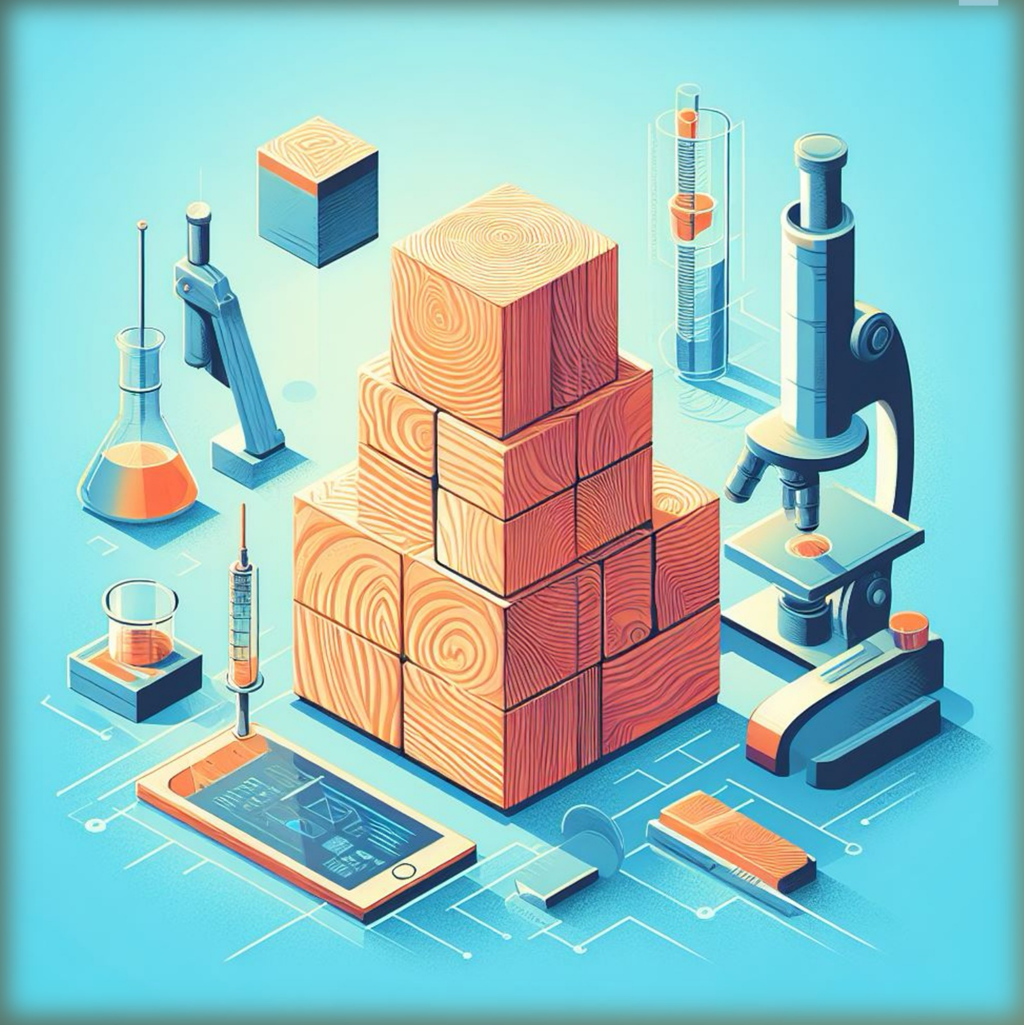
FROM FREQUENCIES TO FLORA

FOURIER ANALYSIS AS A KEY TO SPECIES IDENTIFICATION



OUR TEAM

ALEX | INTRO

CAITLIN | METHOD

DEAN | RESULTS

JANE | DISCUSSION

MORGAN | CONCLUSION

OUR VISION

Identify Materials

Using Fourier analysis, identify a wide range of materials

Analyse Condition

Check the condition of a material using NDT

Distribute

Make this technology accessible to the public

OUR GOALS

Distinguish

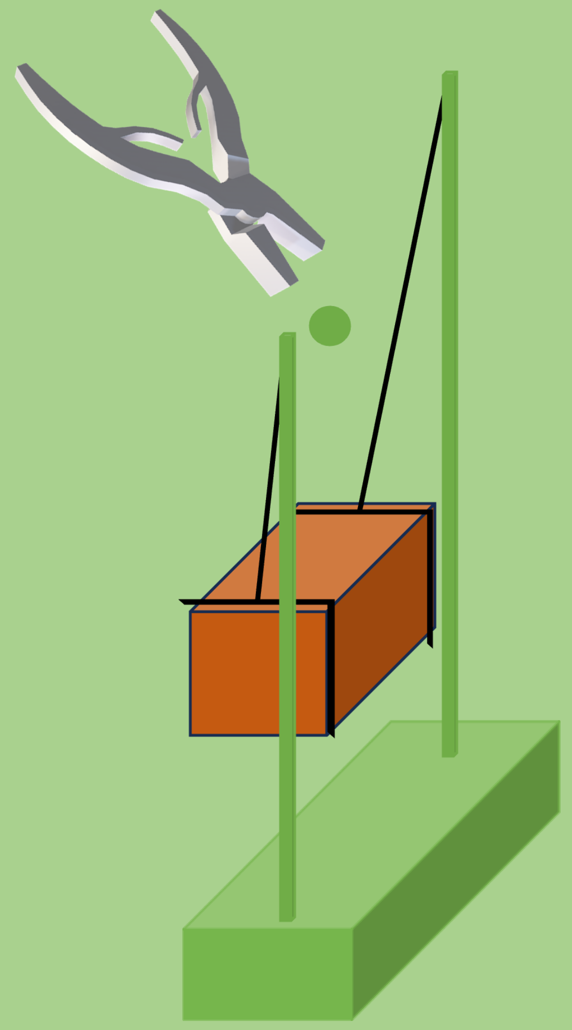
Distinguish between Ash, Oak and Beech blocks based on frequencies produced when struck

Determine

Determine relationship between volume of block and peak frequency

Predict

Use graphic analysis to predict frequencies for any volume of block



METHOD

Block of wood suspended with dimensions:

Largest = ~60mm±1mm x ~60mm±1mm x ~105mm±1mm
Medium= ~60mm±1mm x ~60mm±1mm x ~52.5mm±1mm
Smallest= ~30mm±1mm x ~60mm±1mm x ~52.5mm±1mm

Steel ball dropped

The steel ball weighed 60g ± 0.5g and had a 25mm ± 0 .1mm diameter. Dropped from a height of 80cm±3cm.

Sound recorded

Phone kept a distance 24cm ± 0.1cm from the block and 21cm ± 0.1cm below the block.

METHODS

Graphing

A graph of amplitude in decibels vs time was recorded in the DecibelX app, with an uncertainty of ±0.05 db in amplitude and ±0.05s in time.

Fourier Transform

Python was used to perform a Fourier transform

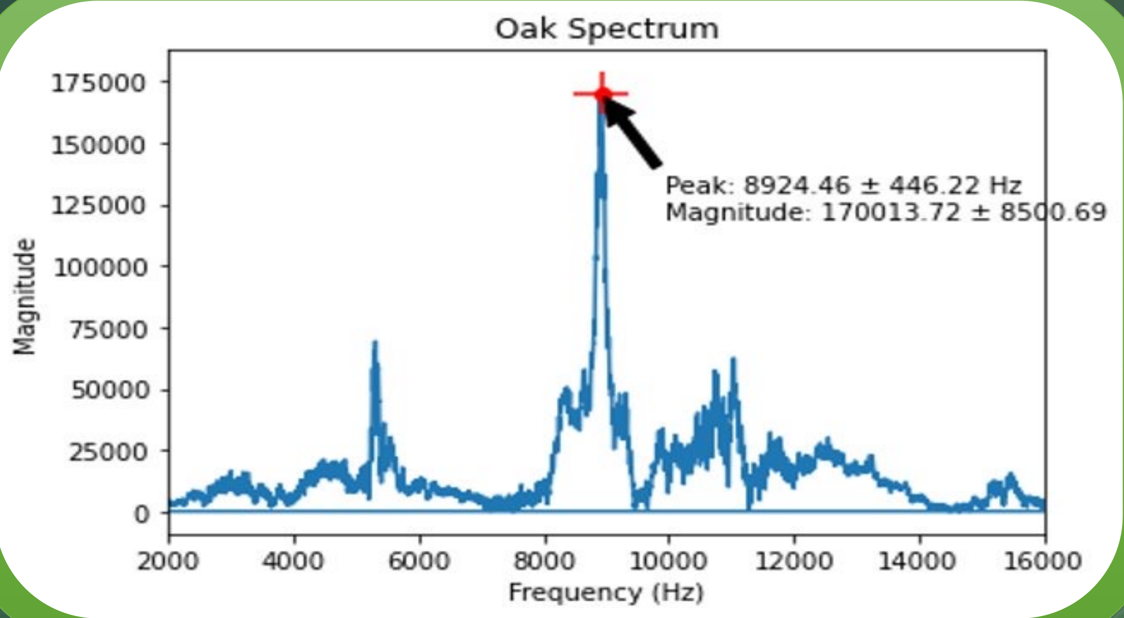
RESULTS- Python

```python
import numpy as np
from scipy.io import wavfile
import matplotlib.pyplot as plt
from scipy.signal import butter, lfilter
```

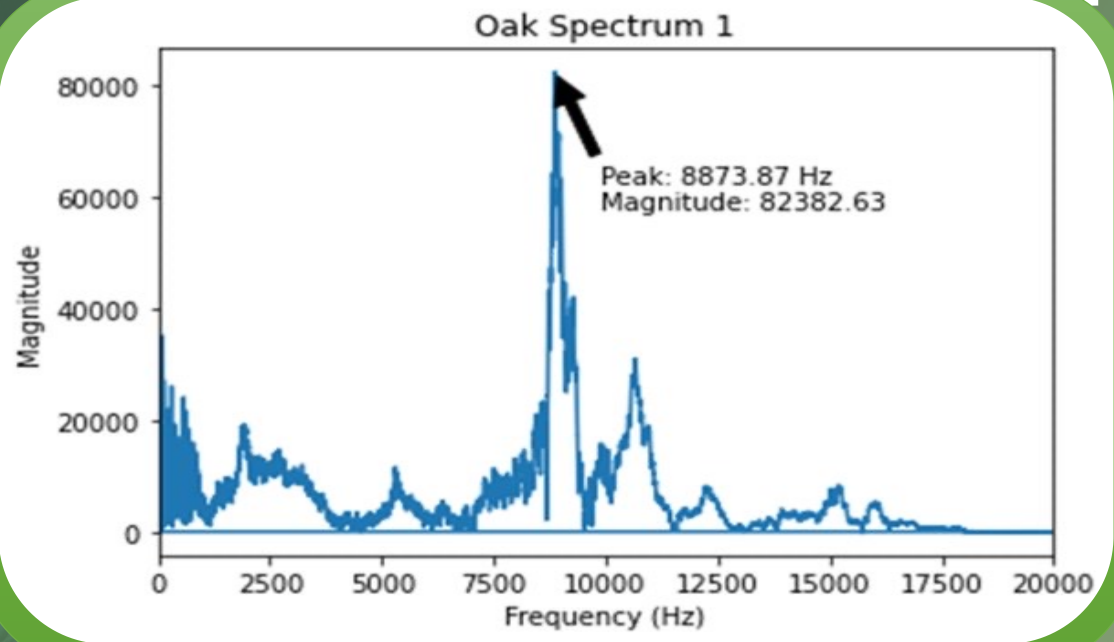
- butter_highpass
- highpass_filter
- remove_background
- compute_fourier_transform
- Plotting

```python
def butter_highpass(cutoff, fs, order=5):
    nyq = 0.5 * fs
    normal_cutoff = cutoff / nyq
    b, a = butter(order, normal_cutoff, btype='high', analog=False)
    return b, a

def highpass_filter(data, cutoff, fs, order=5):
    b, a = butter_highpass(cutoff, fs, order=order)
    y = lfilter(b, a, data)
    return y

def remove_background(data, background_data):
    if len(data) > len(background_data):
        data = data[:len(background_data)]
    else:
        background_data = background_data[:len(data)]
    return data - background_data

def compute_fourier_transform(filename, background_filename=None, trim_start_sec=0):
    rate, data = wavfile.read(filename)

    if background_filename:
        _, background_data = wavfile.read(background_filename)
        data = remove_background(data, background_data)

    if trim_start_sec > 0:
        trim_start_samples = int(trim_start_sec * rate)
        data = data[trim_start_samples:]

    if len(data.shape) == 2:
        data = np.mean(data, axis=1)

    # Remove DC offset
    data = data - np.mean(data)

    # Apply Hamming window
    data = data * np.hamming(len(data))

    # High-pass filtering to remove frequencies
    data = highpass_filter(data, 4000, rate)

    # Compute the Fourier Transform of the audio
    spectrum = np.fft.fft(data)

    freq = np.fft.fftfreq(len(spectrum), d=1/rate)

    return freq, spectrum
```

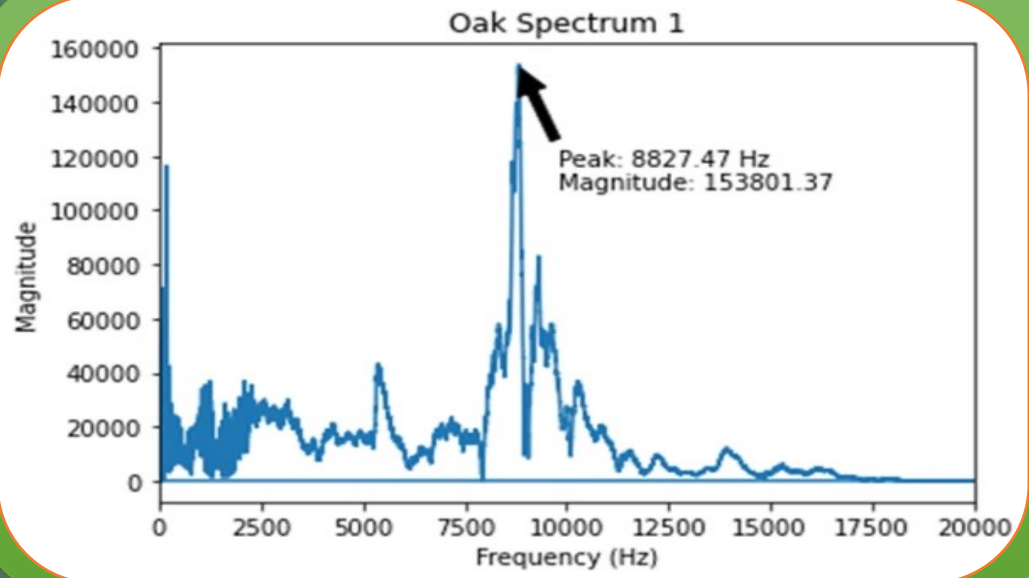
RESULTS



Day 1

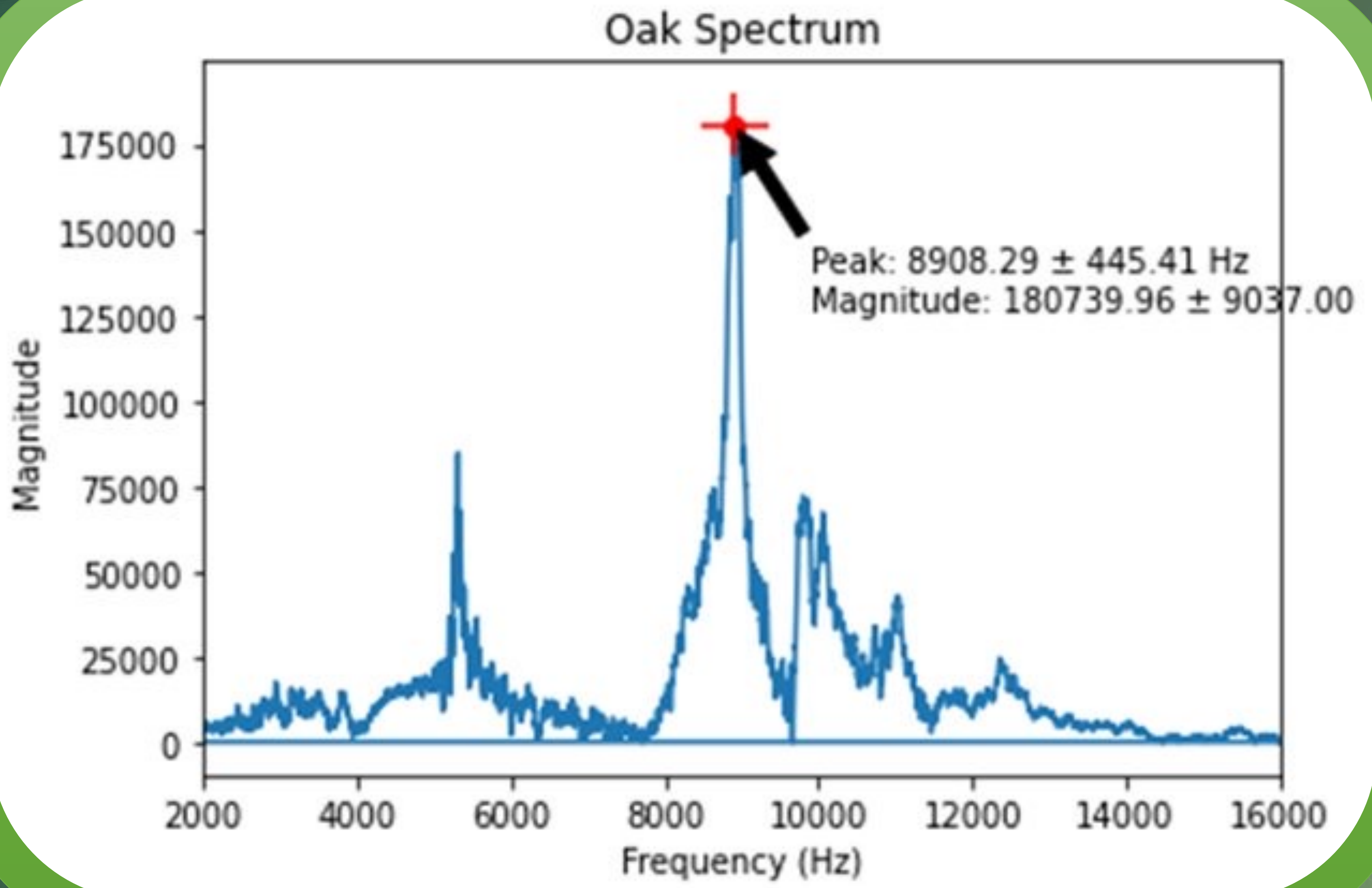


Day 2



Day 3

RESULTS

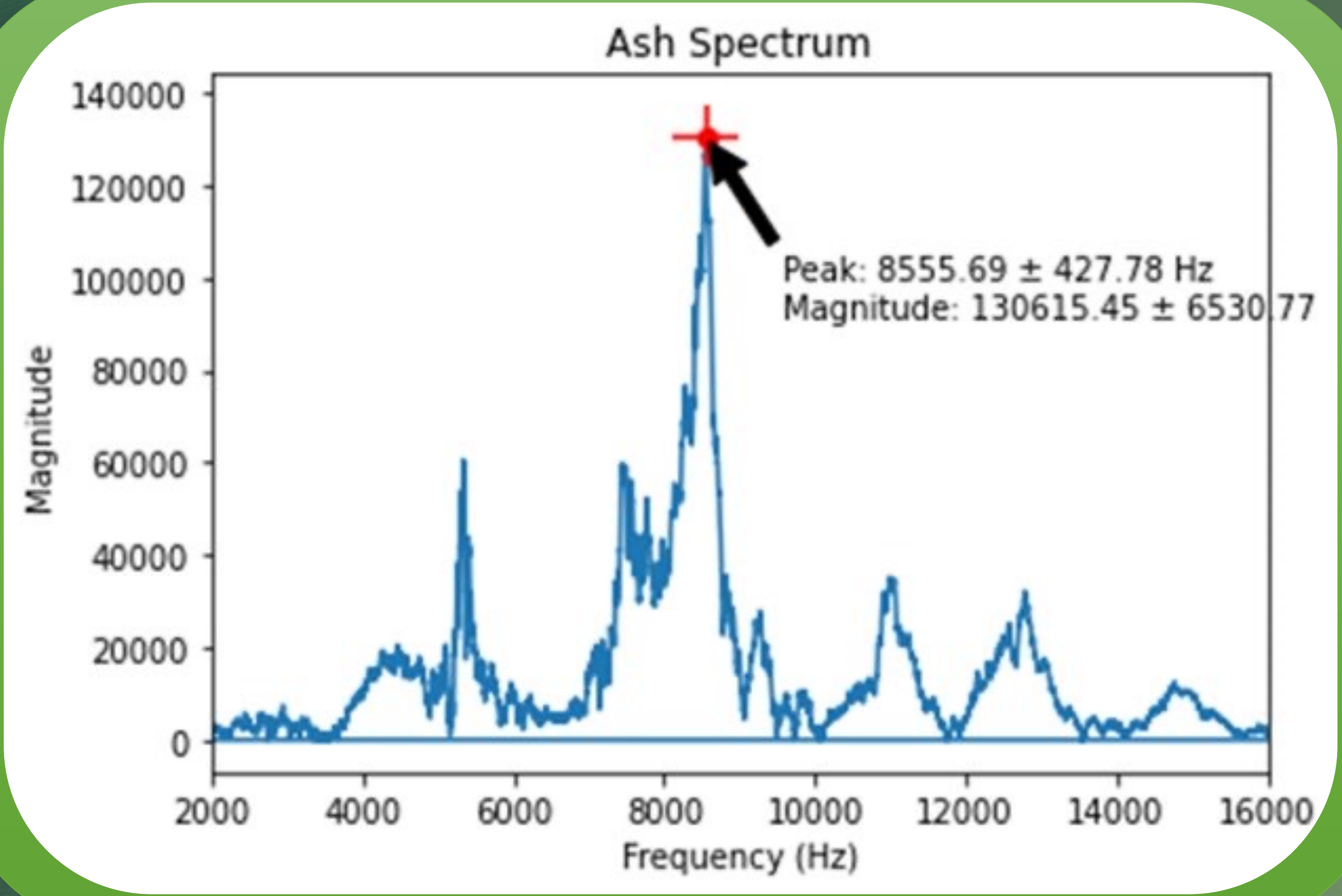


•Nyquist frequency, which was consistently 22050.0 Hz, half the usual sampling rate for WAV files.

Oak: 8908.29 ± 445.41 Hz - Frequency Resolution: 0.4306640625 Hz

Full Blocks

RESULTS

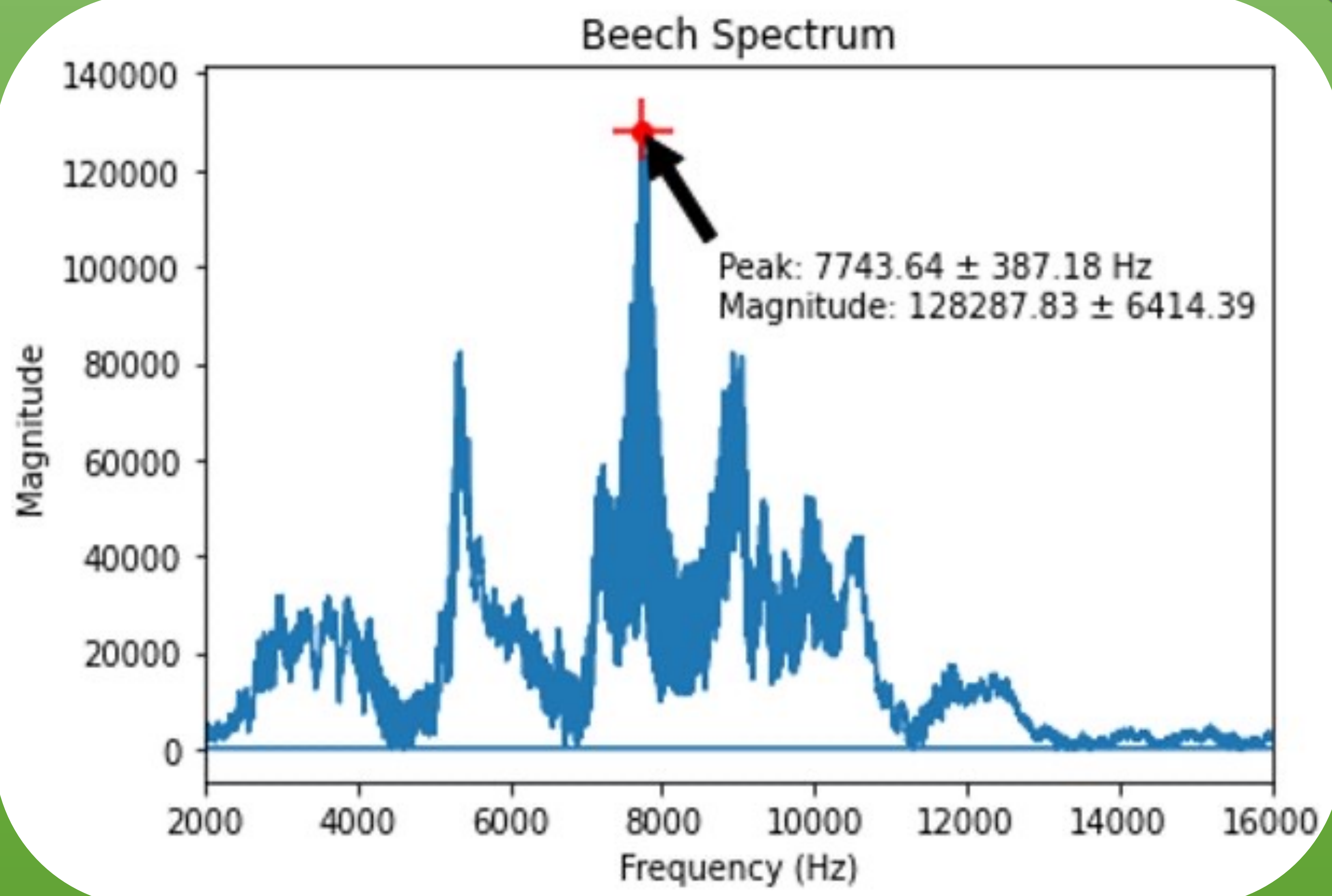


Nyquist frequency, which was consistently 22050.0 Hz, half the usual sampling rate for WAV files.

Ash: 8555.69 ± 427.78 Hz
- Frequency Resolution: 0.566663240131579 Hz

Full Blocks

RESULTS

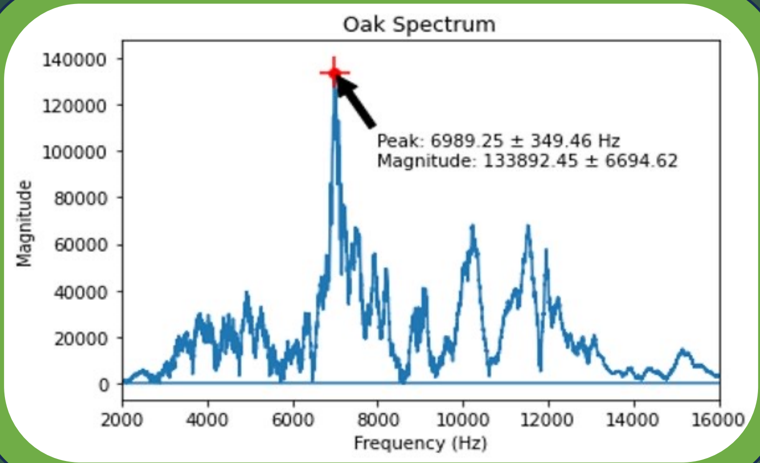


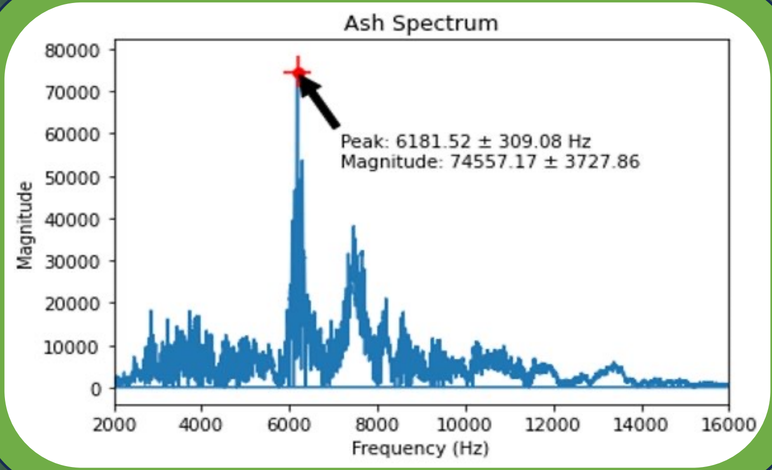
•Nyquist frequency, which was consistently 22050.0 Hz, half the usual sampling rate for WAV files.

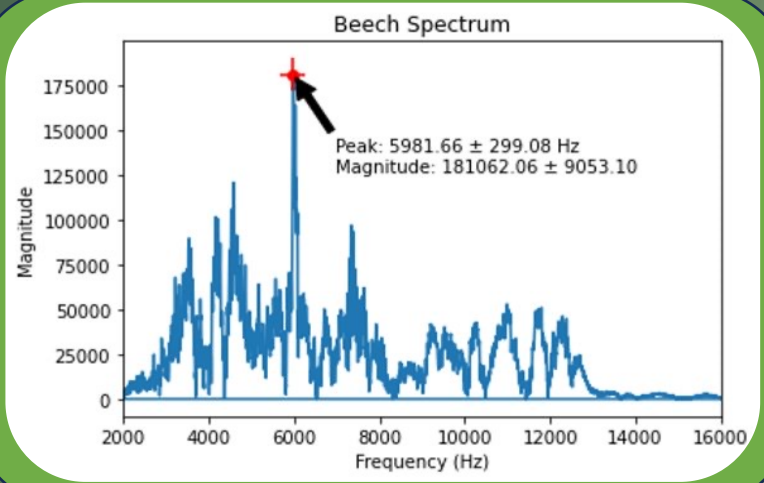
Beech: 7743.64 ± 387.18 Hz - Frequency Resolution: 0.4024897780373832 Hz

Full Blocks

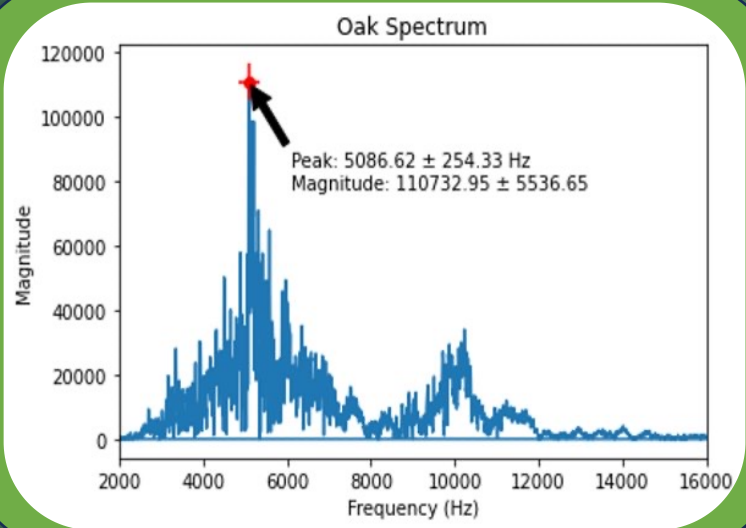
RESULTS

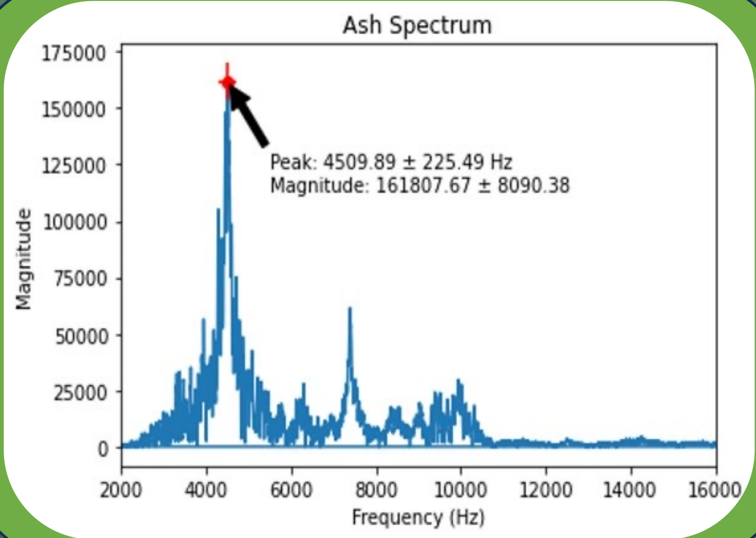


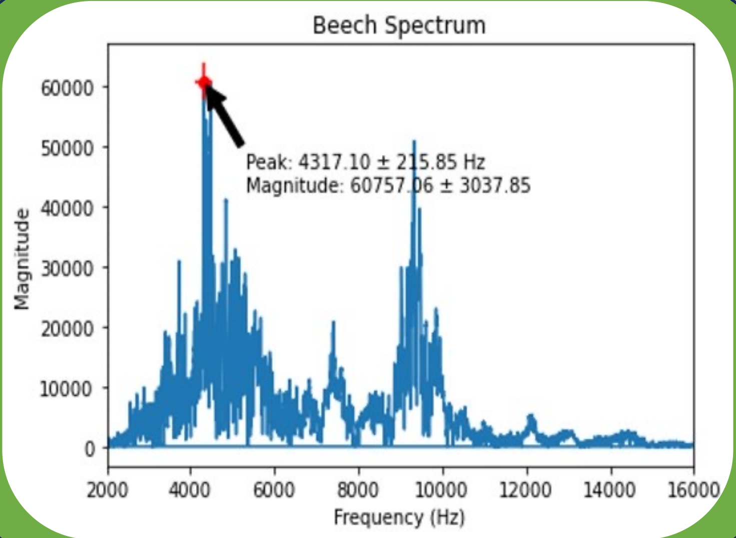




$\frac{1}{2}$ Blocks ↑ ↓ $\frac{1}{4}$ Blocks

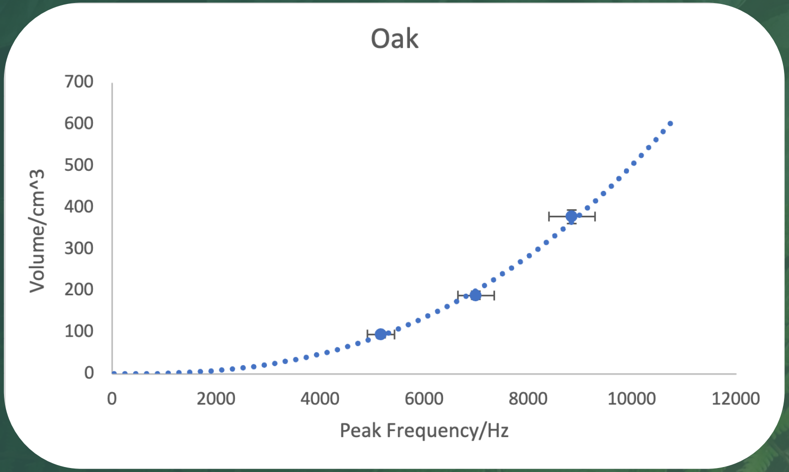


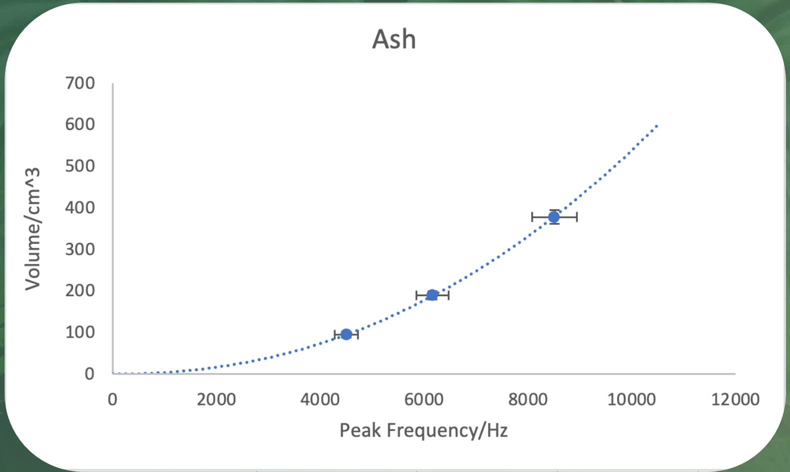


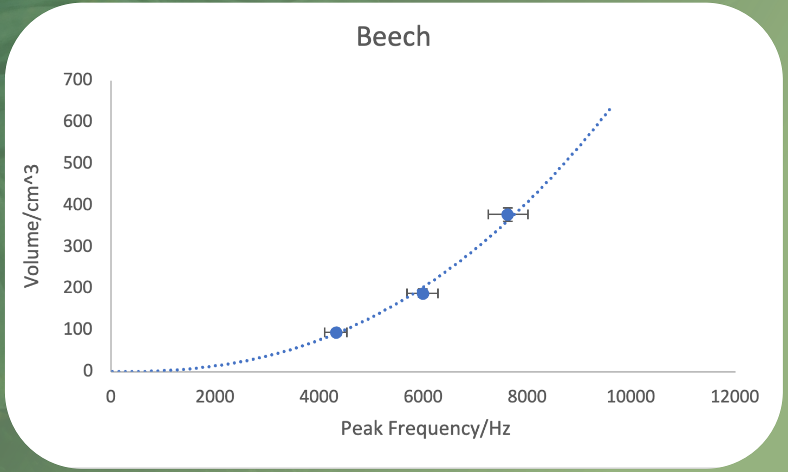


RESULTS

Difference between sizes (Hz)	Type	Full	1/2	1/4
1845.817 1823.62	Oak	8835.215 +- 441.76 Hz	6989.40 +- 349.47 Hz	5165.78 +- 258.289 Hz
2345.747 1662.43	Ash	8510.347 +- 425.517 Hz	6164.60 +- 308.23 Hz	4502.17 +- 225.1085 Hz
1640.903 1668.23	Beech	7626.233 +- 381.411 Hz	5985.33 +- 299.27 Hz	4317.10 +-215.855 Hz







DISCUSSION-RESULTS

Consistency across experiments

Smaller volume, smaller frequency

Differences in materials ascertained

Resonance occurring

FUTURE ASPIRATIONS

Grading Materials Using NDT

Identify Wider Range of Materials Using NDT

FTIR Used to Identify Disease

Ecosystem Protection

CONCLUSION

Material Identification

Demonstrated the ability to use NDT for distinguishing samples.

Frequency Relationship

Exhibited the relationship between block size and frequency.

Material Grading

Demonstrated the ability to use NDT for identifying defaults in samples.

REFERENCES

https://www.bing.com/images/create/infographic-image-of-a-clipboard-with-the-word-con/655128568a544ba5b95618d6e1c7f684?FORM=GENCRE
Images sourced from bing image generator

Information obtained from several studies such as
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9147555/
https://bioresources.cnr.ncsu.edu/resources/vibrational-characteristics-of-four-wood-species-commonly-used-in-wood-products/