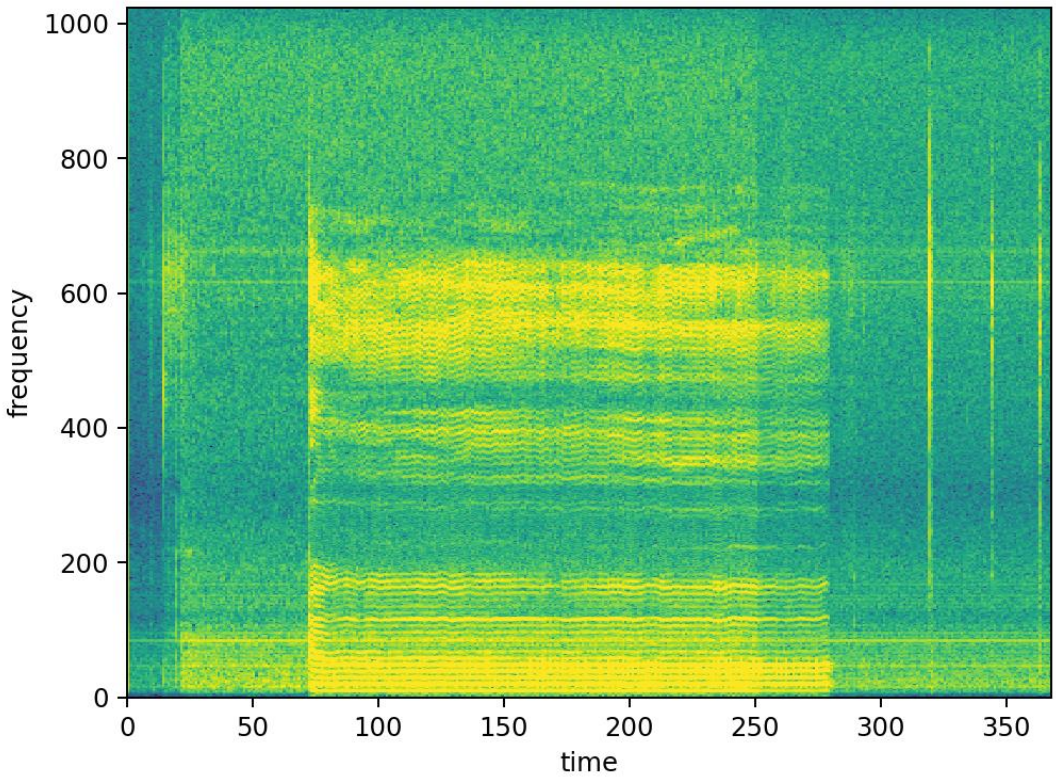
1000
800
600
400
200
0
frequency
0
50
100
150
200
250
300
350
time

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.io.wavfile import read, write
from numpy.fft import fft, ifft
from scipy import signal

FRAME_SIZE = 1024
ZP_FACTOR = 2
FFT_SIZE = FRAME_SIZE * ZP_FACTOR


################### YOUR CODE HERE ######################
def ham_win(N):
    window = []
    for i in range (N):
        data = 0.54 - 0.46*np.cos((2*np.pi*i)/(N-1))
        window.append(data)
    return window



def ece420ProcessFrame(frame):
    window = ham_win(FRAME_SIZE)
    data = frame * window
    for i in range (FRAME_SIZE):
        np.append(data,0)
    data_fft = abs(fft(data,FFT_SIZE))
    output = data_fft * data_fft
    output = np.log(data_fft)/10
    out = output[:FRAME_SIZE]



    return out


################### GIVEN CODE BELOW ######################
```

```
################## GIVEN CODE BELOW ######################

Fs, data = read('test_vector.wav')

numFrames = int(len(data) / FRAME_SIZE)
bmp = np.zeros((numFrames, FRAME_SIZE))

for i in range(numFrames):
    frame = data[i * FRAME_SIZE : (i + 1) * FRAME_SIZE]
    curFft = ece420ProcessFrame(frame)
    bmp[i, :] = curFft

plt.figure()
plt.pcolormesh(bmp.T, vmin=0, vmax=1)
plt.axis('tight')
plt.xlabel('time (ms)')
plt.ylabel('frequency (Hz)')
plt.show()
```