F74062109 許嘉倫 資訊110

程式語言: python

import numpy as np

import librosa

import math

from playsound import playsound

import matplotlib.pyplot as plt

import librosa.display

# test .wav can playsound

filename = '2.wav'

# playsound(filename)

y, sr = librosa.load(filename, sr=None)

y = y/max(abs(y))

# waveform

plt.figure(figsize=(25, 5))

librosa.display.waveplot(y, sr=sr)

plt.title('waveform')

plt.show()

# w(n) = 1/(2\*N) if 0~N-1 else 0

def sgn(arr): # 判斷正負號，正為1，負為-1

for i in range(len(arr)):

if(arr[i]>=0):

arr[i] = 1

else:

arr[i] = -1

return arr

def calZcr(waveData,frameSize,overLap):

wlen = len(waveData)

frameShift = frameSize - overLap

frameNum = math.ceil(wlen/frameShift)

zcr = np.zeros((frameNum,1))

for i in range(frameNum):

curFrame = waveData[np.arange(i\*frameShift,min(i\*frameShift+frameSize,wlen))]

curFrame = sgn(curFrame) #判斷正負號

#代入公式

zcr[i] = sum(abs(curFrame[1:] - curFrame[:-1]))/(2\*frameSize)

return zcr

# 20ms

frameSize = int(0.02\*sr)

overLap = int(frameSize/2)

zcr = calZcr(y,frameSize,overLap)

time\_zcr = np.arange(0, len(zcr)) \* 0.02 \* ((frameSize-overLap)/frameSize)

plt.figure(figsize=(25, 5))

plt.plot(time\_zcr, zcr)

plt.ylabel("ZCR")

plt.xlabel("time (s)")

plt.title('Zero-crossing rate contour')

plt.show()

# w(n) = rectangular window 1 if 0~N-1 else 0

def calEnergy(waveData,frameSize,overLap):

wlen = len(waveData)

frameShift = frameSize - overLap

frameNum = math.ceil(wlen/frameShift)

energy = np.zeros((frameNum,1))

for i in range(frameNum):

curFrame = waveData[np.arange(i\*frameShift,min(i\*frameShift+frameSize,wlen))]

energy[i] = sum(curFrame\*\*2)

return energy

frameSize = int(0.02\*sr)

overLap = int(frameSize/2)

energy = calEnergy(y,frameSize,overLap)

# print(len(zcr))

time\_energy = np.arange(0, len(energy)) \* 0.02 \* ((frameSize-overLap)/frameSize)

plt.figure(figsize=(25, 5))

plt.plot(time\_energy, energy)

plt.ylabel("Energy")

plt.xlabel("time (s)")

plt.title('Energy contour')

plt.show()

# w(n) = 1/(N) if 0~N-1 else 0

def calMagnitude(waveData,frameSize,overLap):

wlen = len(waveData)

frameShift = frameSize - overLap

frameNum = math.ceil(wlen/frameShift)

magnitude = np.zeros((frameNum,1))

for i in range(frameNum):

curFrame = waveData[np.arange(i\*frameShift,min(i\*frameShift+frameSize,wlen))]

magnitude[i] = sum(abs(curFrame))/(frameSize)

return magnitude

frameSize = int(0.02\*sr)

overLap = int(frameSize/2)

magnitude = calMagnitude(y,frameSize,overLap)

time\_magnitude = np.arange(0, len(magnitude)) \* 0.02 \* ((frameSize-overLap)/frameSize)

plt.figure(figsize=(25, 5))

plt.plot(time\_magnitude, magnitude)

plt.ylabel("magnitude")

plt.xlabel("time (s)")

plt.title('Average Magnitude')

plt.show()

# First, center clip, Second autocorrelation, Finally get pitch

# center clip

CL = 0.3\*np.max(abs(y.copy()))

x = np.zeros((len(y),1))

for i in range(len(y)):

if y[i]>=CL:

x[i] = y[i]-CL

elif abs(y[i])<CL:

x[i] = 0

else:

x[i] = y[i]+CL

time\_center\_clip = np.arange(0,len(x))/sr

plt.figure(figsize=(25, 5))

plt.plot(time\_center\_clip, x)

plt.ylabel("center clip speech")

plt.xlabel("time (s)")

plt.title('center clip')

plt.show()

# calculate Rn(k) use autocorrelation

def calPitch(waveData,frameSize,overLap):

R = 0

Rn = 0

maxk = 0

frameShift = frameSize - overLap

wlen = len(waveData)

frameNum = math.ceil(wlen/frameShift)

R = np.zeros((frameNum,1))

for i in range(frameNum):

for k in range(frameShift):

Rn = np.zeros((frameShift,1))

if i\*frameShift+frameSize+k<wlen:

curFrame = x[np.arange(i\*frameShift,min(i\*frameShift+frameSize,wlen))]

nxtFrame = x[np.arange(i\*frameShift+k,min(i\*frameShift+frameSize+k,wlen))]

Rn[k] = sum(curFrame\*nxtFrame)

if abs(Rn[k])>abs(Rn[maxk]):

maxk = k

R[i] = maxk

maxk = 0

return R

frameSize = int(0.02\*sr)

overLap = int(frameSize/2)

pitch = calPitch(x,frameSize,overLap)

time\_pitch = np.arange(0, len(pitch)) \* 0.02 \* ((frameSize-overLap)/frameSize)

plt.figure(figsize=(25, 5))

plt.plot(time\_pitch, pitch)

plt.ylabel("pitch")

plt.xlabel("time (s)")

plt.title('pitch detection')

plt.show()

# 找ITL, ITU, IZCT

def findThreshold(zcr,magnitude):

# find IZCT

IF = 25

zcr\_copy = zcr[:10].copy()

zcr\_mean = np.mean(zcr\_copy)

zcr\_std = np.std(zcr\_copy)

IZCT = min(IF, zcr\_mean+2\*zcr\_std)

# find ITL ITU

magnitude\_copy = magnitude[:10].copy()

IMX = np.max(magnitude\_copy)

IMN = np.min(magnitude\_copy)

I1 = 0.03\*(IMX-IMN)+IMN

I2 = 4\*IMN

ITL = min(I1,I2)

ITU = 5\*ITL

return IZCT,ITL,ITU

#需要用到Average magnitude

def end\_point\_detection(IZCT,ITL,ITU,magnitude,zcr):

ITU\_index = np.where((magnitude>ITU))[0]

ITL\_index = np.where((magnitude<=ITL))[0]

IZCT\_index = np.where((zcr<=IZCT))[0]

start = []

end = []

for i in ITU\_index:

N1 = ITL\_index[np.where((ITL\_index<i))[0][-1]]

if (N1-IZCT\_index[np.where((IZCT\_index<N1))[0][-1]])>=3:

N1 = IZCT\_index[np.where((IZCT\_index<N1))[0][-1]]

if len(start)!=0 and len(end)!=0:

if N1>int(end[-1]):

start.append(N1)

else:

start.append(N1)

N2 = ITL\_index[np.where((ITL\_index>i))[0][0]]

if (N2-IZCT\_index[np.where((IZCT\_index>N2))[0][0]])>=3:

N2 = IZCT\_index[np.where((IZCT\_index>N2))[0][0]]

if len(start)-len(end)==1:

end.append(N2)

return start,end

IZCT,ITL,ITU = findThreshold(zcr,magnitude)

start, end = end\_point\_detection(IZCT,ITL,ITU,magnitude,zcr)

plt.figure(figsize=(25, 5))

librosa.display.waveplot(y, sr=sr)

plt.title('waveform')

for s, e in zip(start, end):

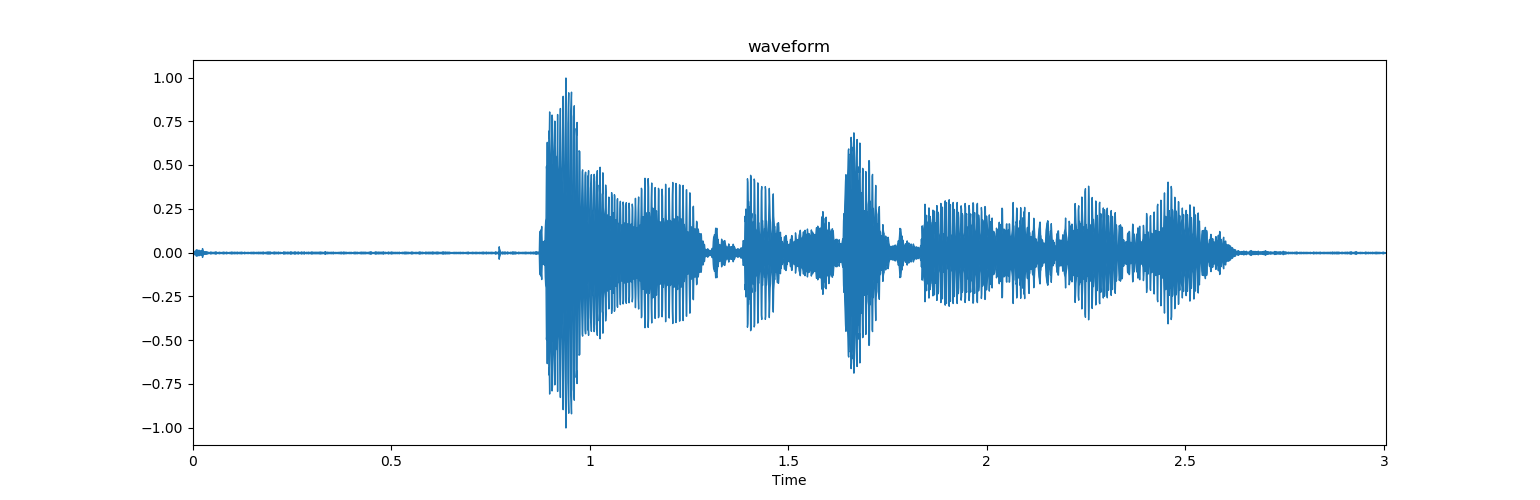
plt.axvline(x=s\*0.02, color='green')

plt.axvline(x=e\*0.02, color='red')

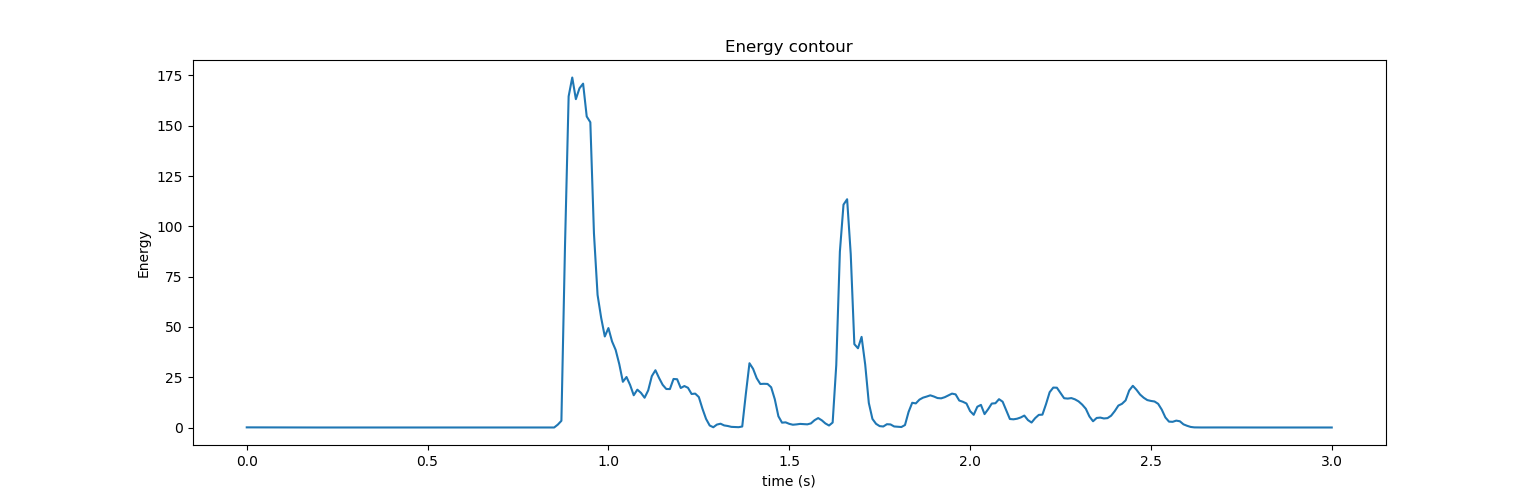
plt.show()

程式碼說明:

1.Waveform



2.Energy contour



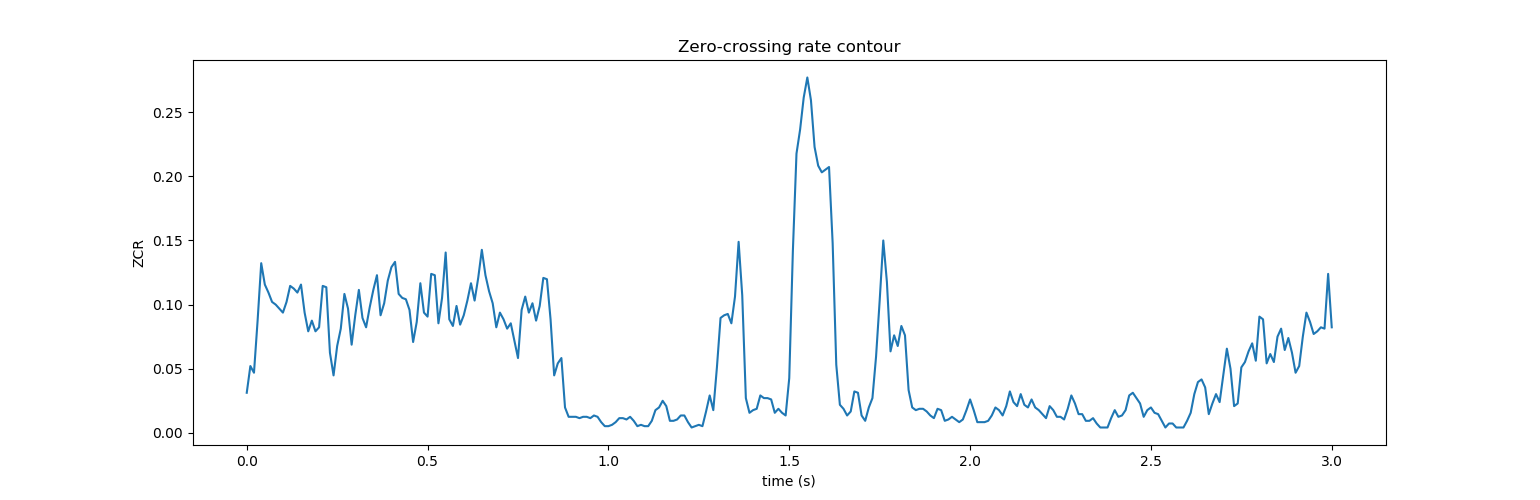
這裡是用short-time Energy

frameSize取20ms, overLap取frameSize/2, 所以frameShift就是frameSize-overLap

window function是rectangular



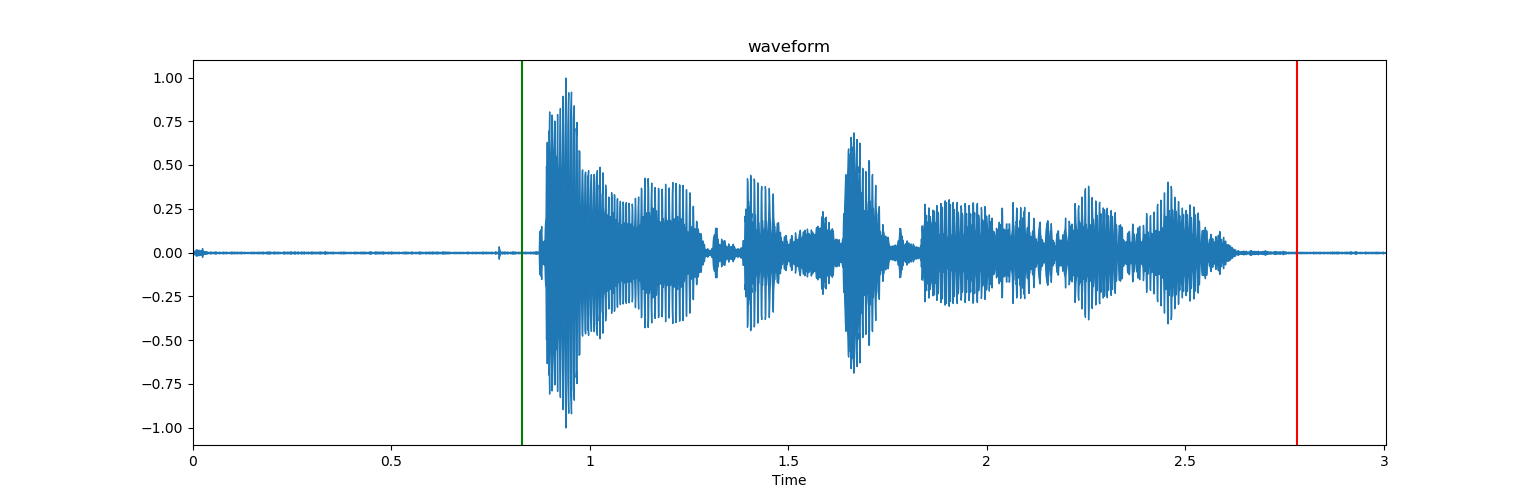
3.Zero-crossing-rate



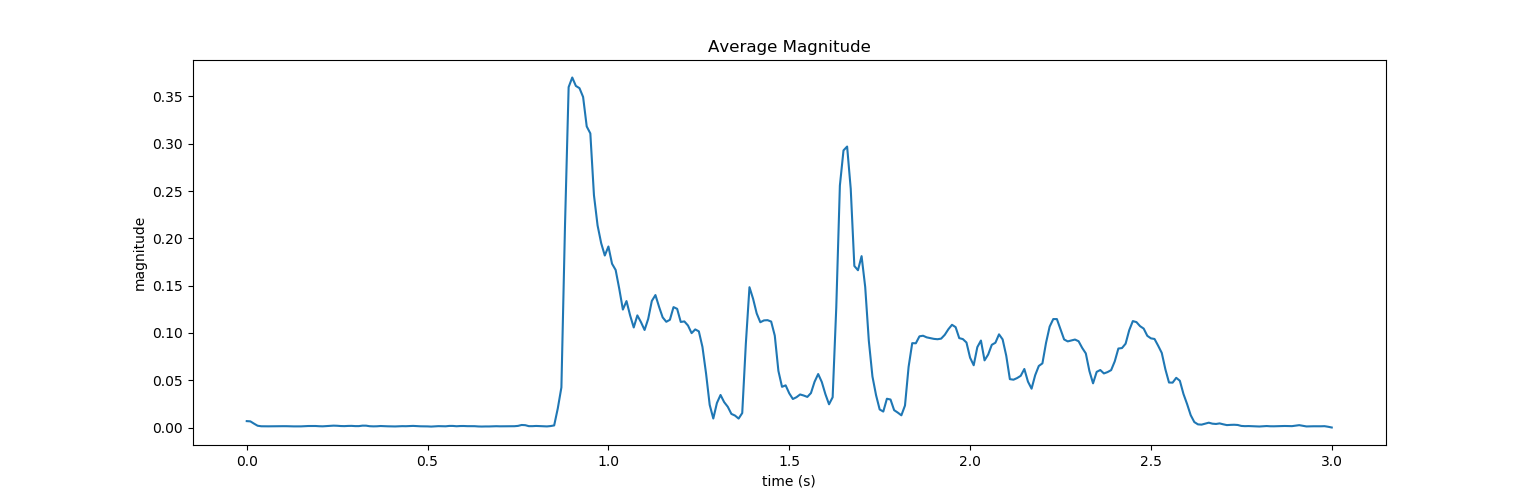
這裡要先判斷正負，正為1，負為-1 , sgn就是判斷的function



4.End-point-detection



首先要先算Average magnitude和Zero-crossing-rate

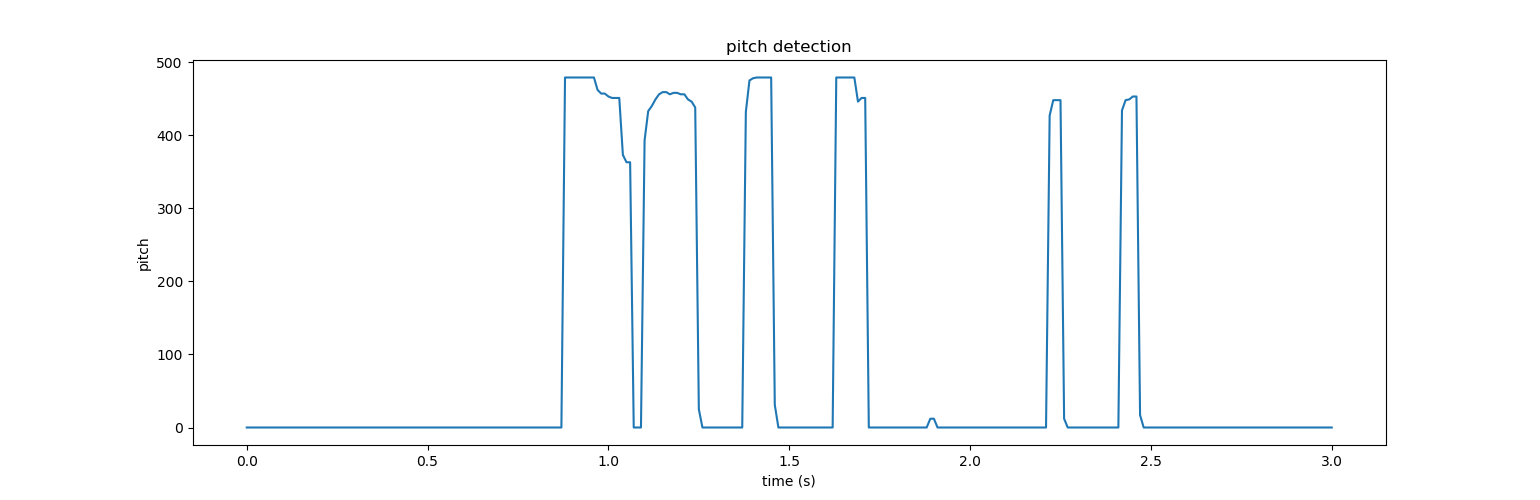


再找出ITU, ITL, IZCT公式為

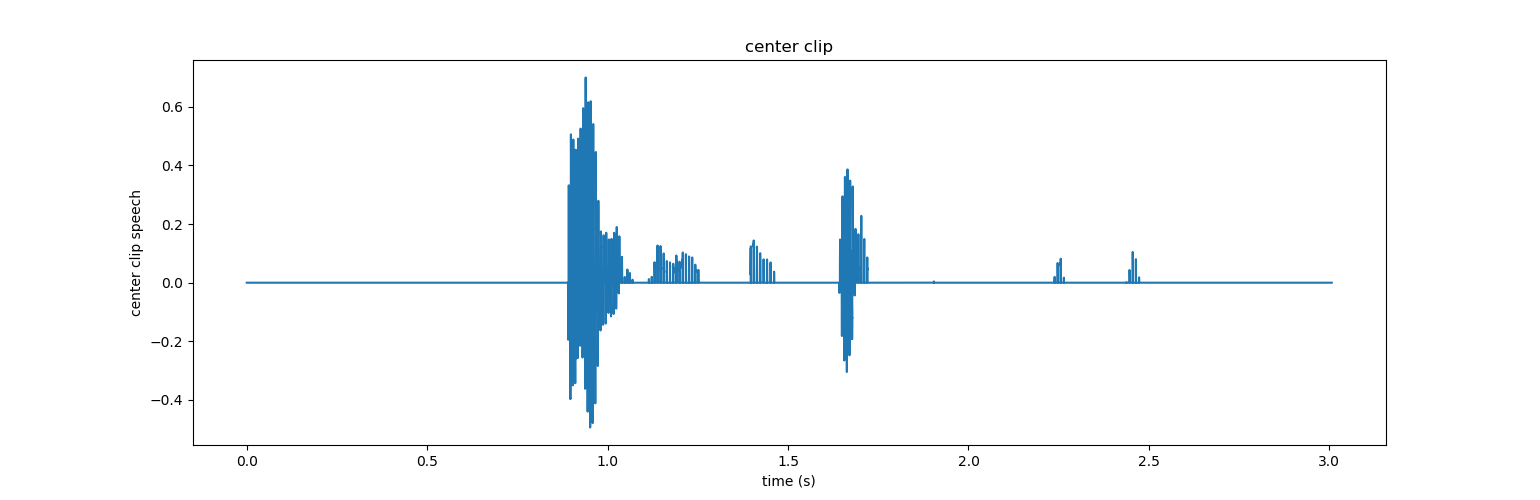
 

接著再按照講義的演算法就可以找出End point

5. pitch detection



首先先做center clip如下圖

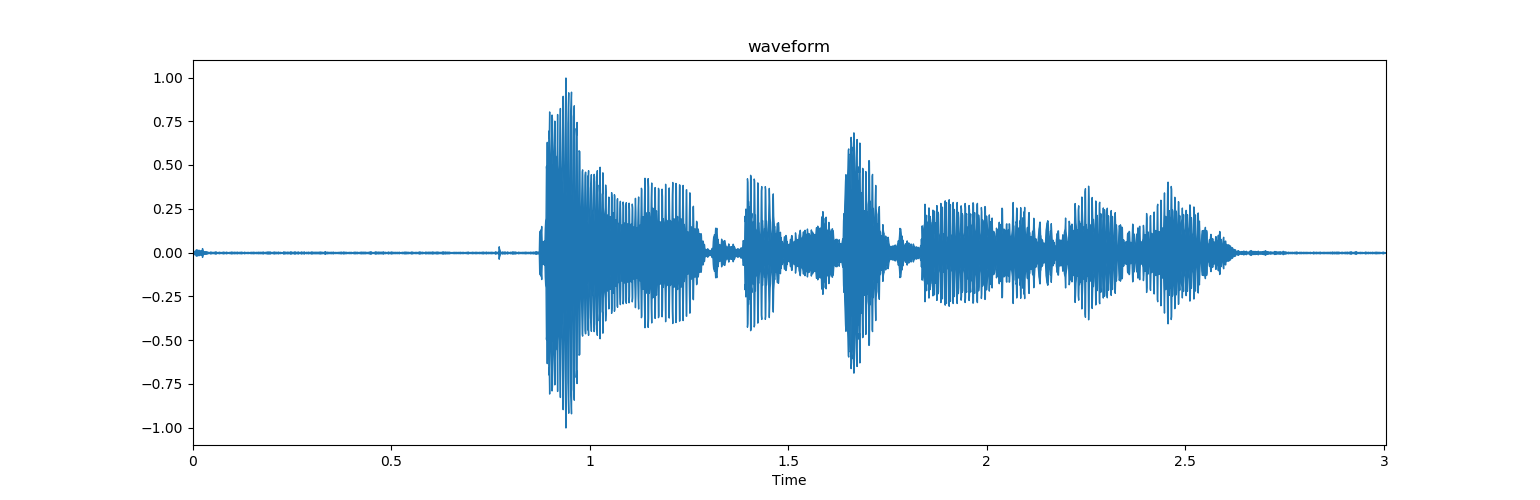


再做autocorrelation就可以得到pitch

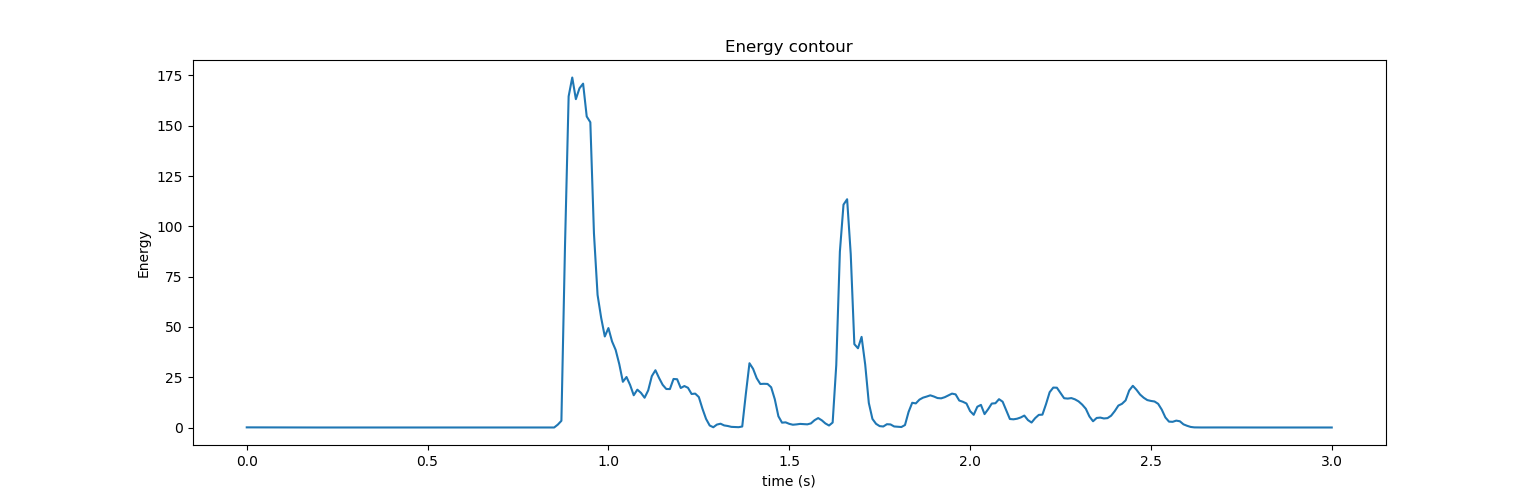
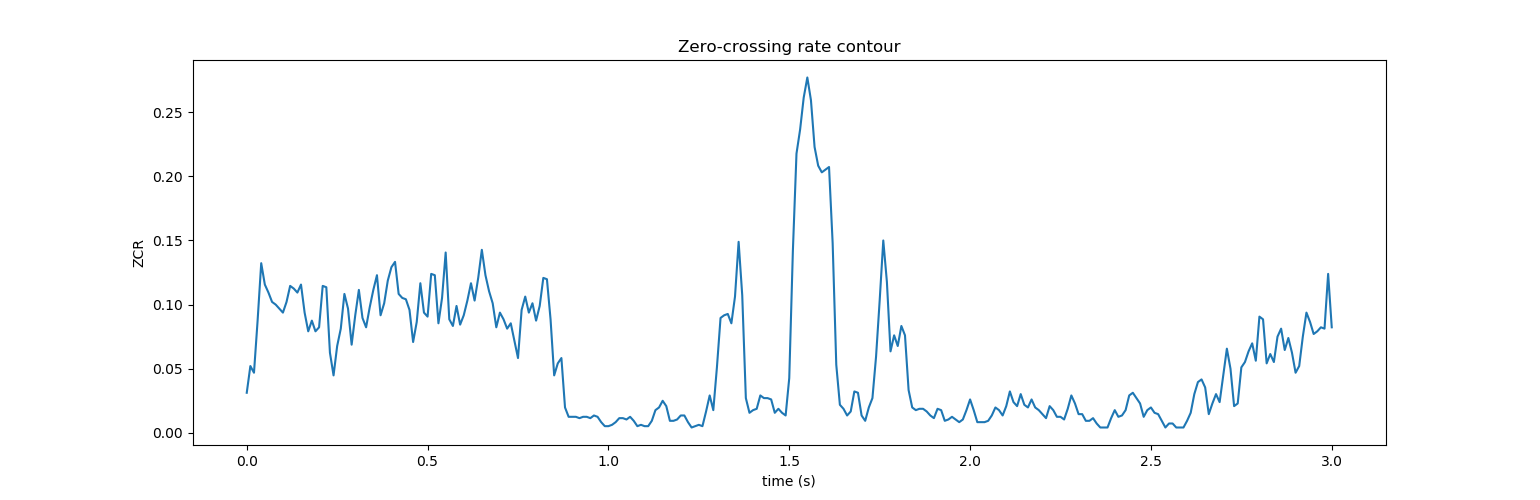


輸出結果:

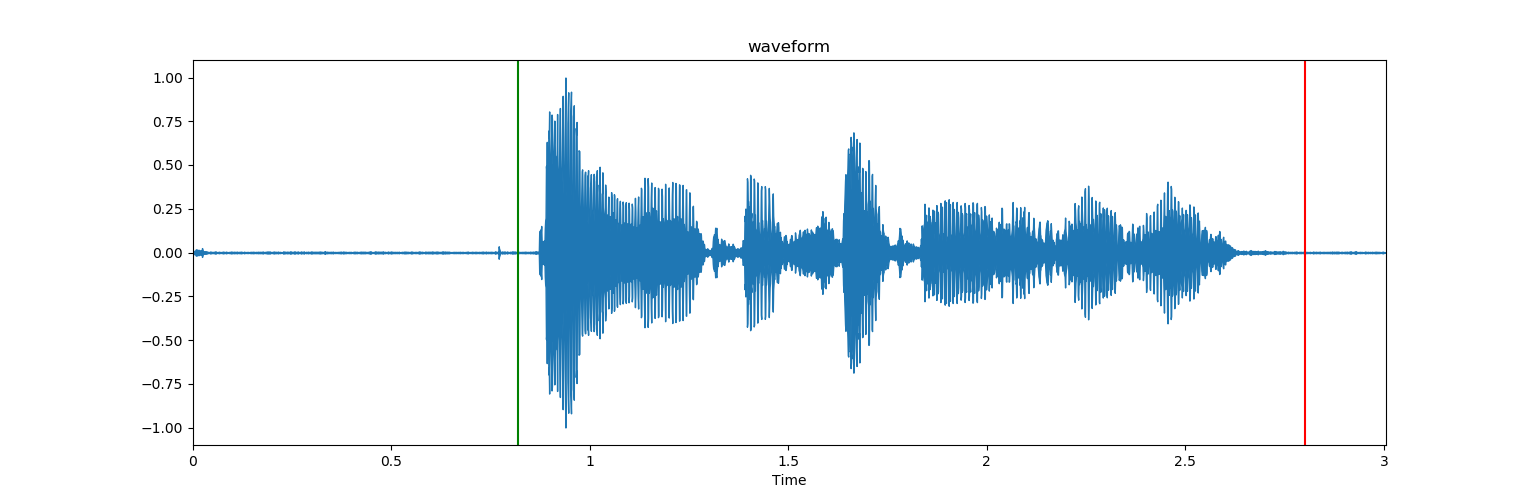
waveform



Energy contour

zero crossing rate

End point detection



Pitch detection

