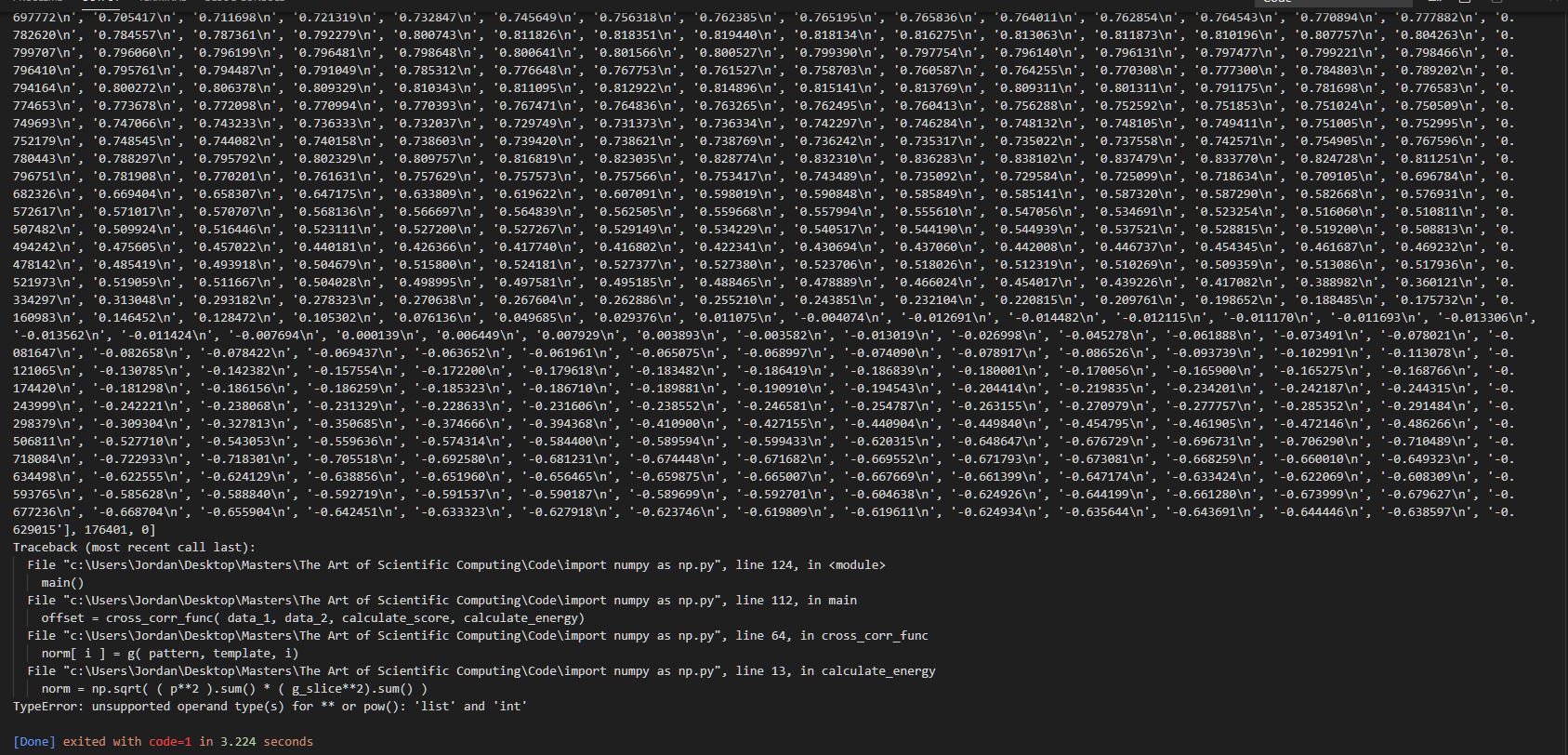
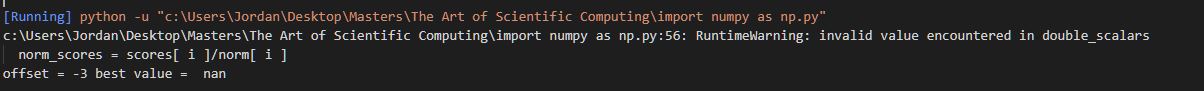
Appendix

**Debugging**

* Error reading noise data
  + Using try:except and print all of the lines. found htat noise data is text not a list bc has 'enter return'  so need to process the data in a way that is specific to what is being read
  + So insted stor in a numbpy array and then stip on the numerical values by ensuring line variable does not have ny formatting vlaues
  + this was verified using boolen 'isnumeric'  stripped carriage return '\n' and conveted teh reult to float, and skipped th header. So was effiecient to slice that line out. which allows for a general line strip that worked.



* line 49 CCR
  + using deubgging of VS
  + using import pdb (before cross corr)
  + found either pattern or norm are zero since returning nan
  + so bad if overlap are all zeros
  + Fix: Use if norm == 0
  + Beucase when pading would get temaplte slice at begining and end that =0 and so it was possible to get 0 entries that would make the function return Nan.
  + So only need to compute normal if sum!=0



def explicit\_correlation(image, kernel): This is O(n^4)

    hi, wi= image.shape

    hk, wk = kernel.shape

    image\_padded = np.zeros(shape=(hi + hk - 1, wi + wk - 1))

    image\_padded[hk//2:-hk//2, wk//2:-wk//2] = image

    out = np.zeros(shape=image.shape)

    for row in range(hi):

        for col in range(wi):

            for i in range(hk):

                for j in range(wk):

                    out[row, col] += image\_padded[row + i, col + j]\*kernel[i, j]

    return out

* This is for R G B colour images, so will take FFT three times. Above algorithm is for a single channel.