```
import numpy as np
import matplotlib.pyplot as plt
from scipy import misc
import os


class Image(object):
    """
     Class to fix a .pgm image with randomly shifted lines.
    :param fileName is the path to a .pgm file
    """

    image = 0.
    cols = 0
    rows = 0

    threshold_top = 0.995 # Maximum value of normalised cross correlation to
consider shifting.
    threshold_bottom = 0.8 # Minimum value of normalised cross correlation to
consider shifting.

    def __init__(self, fileName):

        self.fileName = str(fileName)
        self.image = misc.imread(self.fileName)
        self.cols = len(self.image)
        self.rows = len(self.image[0])

    def get_xcorrelation(self, line_number):
        """
        :param line_number: int from 1 to rows
        :return: real part of cross correlation between line #line_number and line
#line_number-1
        """

        base_line = np.fft.fft(self.image[line_number])
        base_line[:] = (base_line[:]-np.average(base_line))/(np.std(base_line))
        compared_line = np.fft.fft(np.roll(self.image, 1, axis=0)[line_number])
        compared_line[:] = (compared_line[:] - np.average(compared_line)) / (np.std
(compared_line))
        xcorrelation = np.fft.ifft(np.multiply(base_line, np.conjugate
(compared_line)))

        return xcorrelation.real

    def shift(self):
        """
        Use get_xcorrelation() function to shift the input image into place.
        """

        xcorrelation_max = np.zeros(self.cols)
        shift = np.zeros(self.cols, dtype=int)

        for i in range(1, self.cols):

            xcorrelation = self.get_xcorrelation(i)
            shift[i] = (int)(np.argmax(xcorrelation))
            xcorrelation_max[i] = xcorrelation[shift[i]]

            if (shift[i]!=0) & (shift[i] != shift[i - 1]) & \
                    (xcorrelation_max[i] < self.threshold_top) & (xcorrelation_max
[i] > self.threshold_bottom):
                """
                If the shift is non zero, and different from the previous shift
(to avoid bulk shifting),
                there will be a shift provided the cross correlation maximum is
within an appropriate range.
                """
                self.image[i] = np.roll(self.image[i], -shift[i])
```

```
    def clean_up(self):
        """
        Proposed function for cleaning up edges.
        Not used
        """
        cleaned_lines = 0
        xcorrelation_argmax = np.zeros(self.rows)

        for i1 in range(1, self.cols-1):

            xcorrelation_argmax[i1] = np.argmax(self.get_xcorrelation(i1)).astype
(int)

        for i1 in range(1, self.cols-1):

            if (xcorrelation_argmax[i1] != 0):

                self.image[i1] = np.roll(self.image, 1, axis=0)[i1]
                cleaned_lines += 1

    def show_image(self):

        plt.imshow(self.image, cmap=plt.cm.gray)
        plt.show(block=False)

    def save_image(self):

        if os.path.isdir("img_out") == False:
            os.mkdir("img_out")

        out_image = str("img_out/" + input("Output image name (just name, not the
path) : "))
        misc.imsave(out_image, self.image)
```