Image Restoration

Naman Jain - 16110100

This is a simple implementation of a Wiener filter to restore an image with blur and noise.

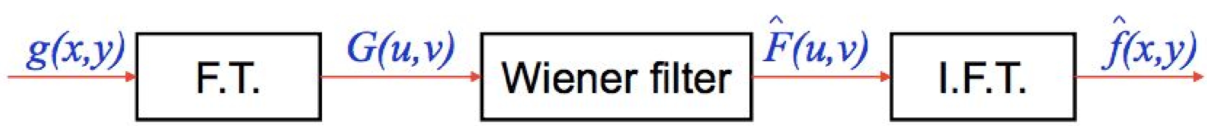
Training the model -

I started off by taking an image of Lena and added gaussian blur along with gaussian noise.

- Gaussian Blur - A kernel of size L x L is taken with all the elements as zero and then the middle element is made 1. Gaussian distribution is applied on kernel which will convolve with the Image to add blur.

- Gaussian Noise - A kernel of the size of the Image gets gaussian distribution with sigma and mean. It is then added to the blur image and is hence correlated.

After getting the Image, the process starts:



g(x,y) - Input Image with noise and blur

G(x,y) - Fourier Transform of g(x,y)

F_rest(x,y) - Output from Weiner filter

f_rest(x,y) - Inverse Fourier transform of F_rest(x,y)

$$G(x,y) = H(x,y)F(x,y) + N(x,y)$$

$$F\_rest(x,y) = W(x,y)G(x,y)$$

Here,

H(x,y) - FT of h(x,y) [Blur Kernel]

N(x,y) - FT of n(x,y) [Gaussian Noise]

$$W(x,y) = H^*(x,y) / (H(x,y)H^*(x,y) + K(x,y))$$

Here,

$K(x,y) = S_N(x,y) / S_O(x,y)$ = Signal to Noise Ratio

$H^*(x,y)$ = Conjugate matrix of H(x,y)

This ratio is taken to be constant while designing the filter in order to eliminate the requirement of original image.

The steps described are followed to get the filter with a specified K and H(x,y).

I trained the model on Lena with Gaussian Blur [Mean=0, Var=1, L=5] and Gaussian Noise [Mean=0, Var=100].

Gray Scale Image

Blur + Noisy Image (g(x,y))

The Noisy image is multiplied by Weiner filter which is build with H(x,y) of blur kernel and different values of K.

The restored image is obtained by the Inverse fourier transform of the product.

Result: The following are the restored images -

K = 0.15

MSE = 101.49

PSNR = 28.06

K = 0.09

MSE = 92.64

PSNR = 28.46

K = 0.03

MSE = 86.5

PSNR = 28.75

Also,

For Gaussian Blur [Mean=0, Var=2, L=2] and Gaussian Noise [Mean=0, Var=50]

Blur + Noisy Image (g(x,y))

K = 0.15

MSE = 101.84

PSNR = 28.05

K = 0.09

MSE = 90.40

PSNR = 28.56

K = 0.03

MSE = 70.43

PSNR = 29.65

Metrics: (w.r.t. the original clear image)

MSE = Mean Square Error

PSNR = Peak to Noise Signal Ratio

K values are taken such that PSNR is least at 0.15 and MSE is least at 0.03.

Using the Code:

```
>> python weiner.py train [IMG_NAME] [BLUR_KERN_SIZE] [VAR_BLUR] [VAR_NOISE] [K]
```

--- Put the input image in the “Weiner” folder and output will come in “test” folder

IMG_NAME --- Image name

BLUR_KERN_SIZE --- Kernel size of blur

VAR_BLUR ---- Variance for Gaussian Blur

VAR_NOISE ---- Variance for Gaussian Noise

K --- K value

Testing the model -

```
>> python weiner.py test [IMG_NAME] [BLUR_KERN_SIZE] [VAR_BLUR] [VAR_NOISE]
```

You can use any square image and add any blur and noise. The algorithm will try to make the image less blurry and less noisy.

The train run saves the filter features such as H and K. This generates the same Wiener filter used for Lena before.

For a car image:

Noisy Image

Restored Image

MSE: 91.81

PSNR: 28.50

References:

- http://www.ee.columbia.edu/~xlx/ee4830/notes/lec7.pdf
- http://www.cis.rit.edu/class/simg782/lectures/lecture_16/lec782_05_16.pdf
- http://www.ee.columbia.edu/~sfchang/course/dip-S04/sample/wiener-filter.htm
- https://lib.dr.iastate.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=3004&context=qnde
- https://researchweb.iiit.ac.in/~aabhas.majumdar/mywebsite/ImageRes.pdf
- http://www.owlnet.rice.edu/~elec539/Projects99/BACH/proj2/wiener.html
- http://www.robots.ox.ac.uk/~az/lectures/ia/lect3.pdf