Problem1

The image “motion blur_1_3024x4032.jpg” is corrupted by motion blur, as illustrated in the left figure. We do not have any information of the motion. From the figure, however, you can tell that different parts of the keyboard seem to have different degrees of blur, perhaps because the camera used to capture the image is at a slant angle with respect to the keyboard. You are asked to restore the image by all means and apply the same method to the toy image on the right, “Donald_Duck.jpeg.”

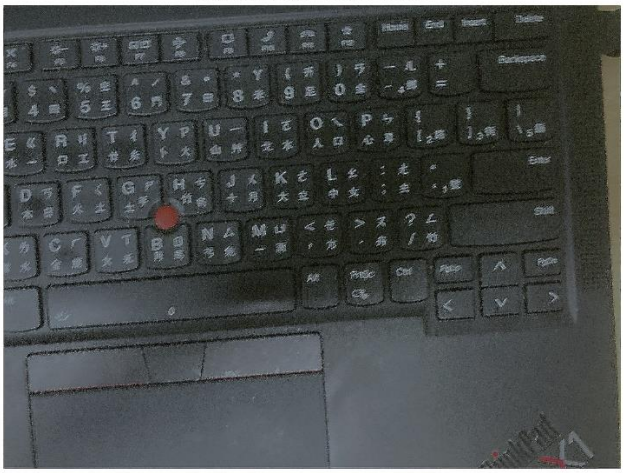
我這次實作了 4 個常見的 de-blurring method，並 based on 它們再加以改良。兩張圖使用的方法都是一樣的，只有 motion function 和一些參數不同。為了讓程式跑快一點，我只執行我認為表現最好的，剩下的都在註解，可自行修改。

Method1: Wiener filtering

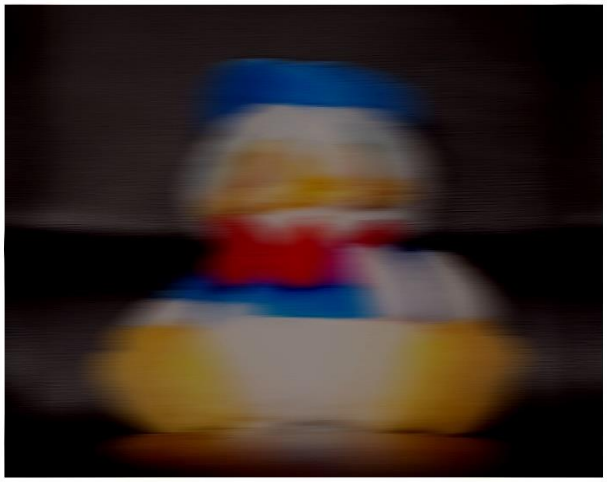
檔名: key_wiener_filter.jpg 和 duck_wiener_filter.jpg

結果:

keyboard_wiener_filter



duck_wiener_filter



說明:

依照課本 p360 Eq 5-85 的公式進行化簡:

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right]G(u,v) = \left[\frac{H^*(u,v)}{H^*(u,v)\,H(u,v) + K}\right]G(u,v)$$

Step1: 用這個公式去 deblur，取各自表現最好的K。

Step2: Apply new filter (出自參考資料[4])，在這些傳統的 deblurring method 後使用這個 new filter，可以提高 PSNR，讓結果更好。

Step3: 做一些色彩、亮度、對比度的調整，使其更貼近原始的 input。

Keyboard 整體看起來有變比較清楚。然後鴨子的殘影變小，遠看比原圖更有一隻鴨子的形狀(原圖看起來像兩隻鴨子)。

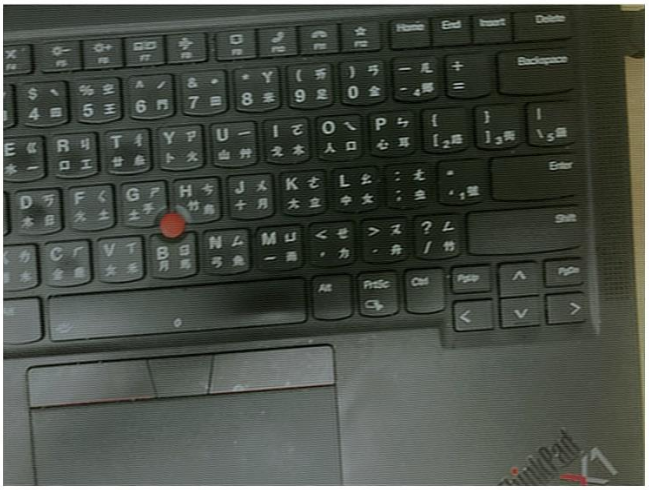
我認為我的鴨子在這個方法表現最好

Method2: Inverse filtering

檔名: key_inverse_filter.jpg 和 duck_inverse_filter.jpg

結果:

keyboard_inverse_filter



duck_inverse_filter

說明:

Step1: 使用課本 p356 Eq 5-78 $\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$ ，為了避免下一頁說的 encountering zero values，所以我有加上一個 butterworth low pass filter (一開始是試 ideal LPF 但發現 ringing 現象太嚴重了)，達到 limiting the analysis to frequencies near the origin。

Step2、Step3: 同方法一。

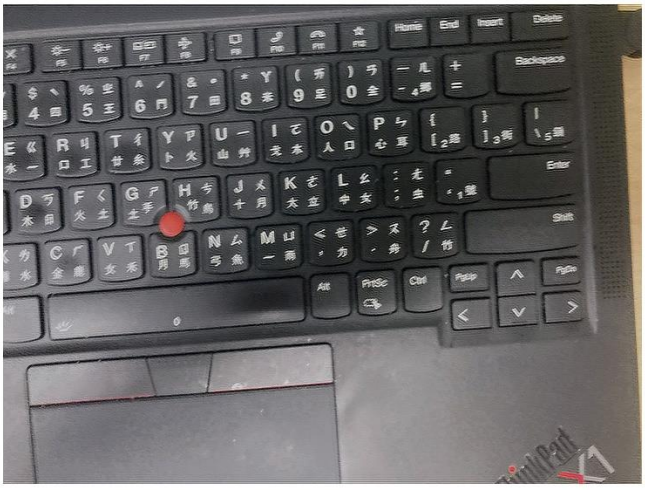
雖然結果相較其他方法是模糊些，但經過 inverse filter 的 keyboard 完全看不出來有 motion blur。鴨子遠看也是比原圖還要更像完整的一隻。

Method3: constrained least squares filtering

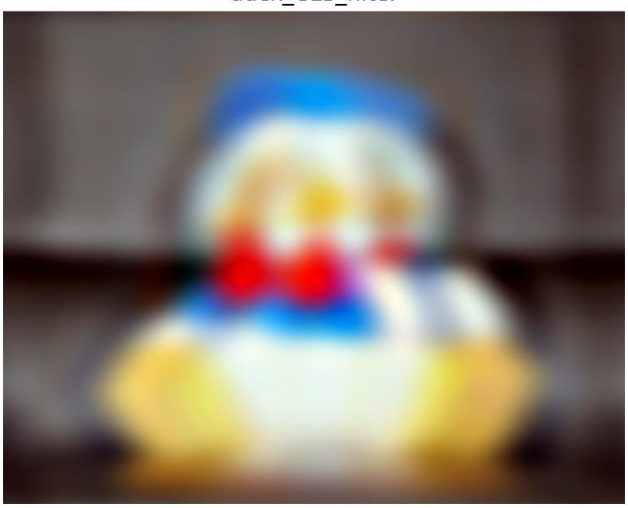
檔名: key_CLS_filter.jpg 和 duck_CLS_filter.jpg

結果:

keyboard_CLS_filter



duck_CLS_filter



說明:

Step1: 使用課本 p364 Eq5-89 $\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2+\gamma|P(u,v)|^2}\right] G(u,v) = \left[\frac{H^*(u,v)}{H^*(u,v)H(u,v)+\gamma|P(u,v)|^2}\right] G(u,v)$

其中 $p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ 。

Step2: 做一些色彩、亮度、對比度的調整，使其更貼近原始的 input。

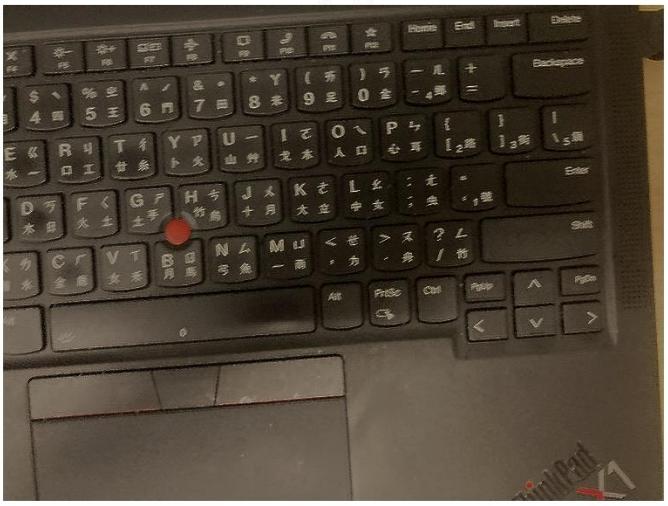
Keyboard 整體看起來有變比較清楚。然後鴨子的殘影變小，遠看比原圖更有一隻鴨子的形狀(原圖看起來像兩隻鴨子)。

Method4: Richardson–Lucy deconvolution

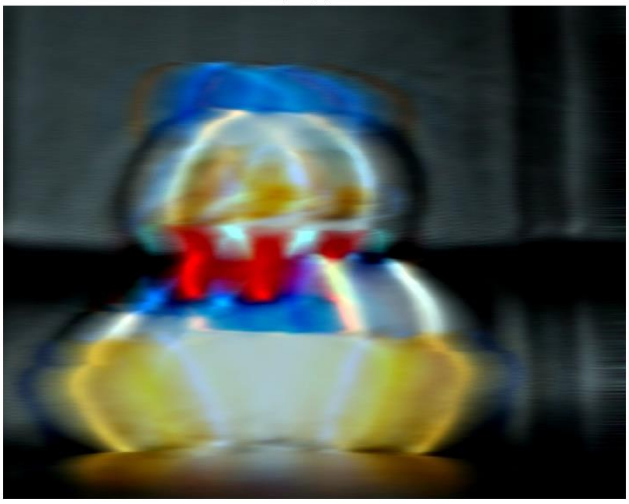
檔名: key_RL_deconv.jpg 和 duck_RL_deconv.jpg

結果:

keyboard_RL_deconv



duck_RL_deconv



RL deconvolution 是一種用 iterative 的方法去 de-blur，實作步驟如參考資料[14]。
大多情況是 iterate 越多次出來的效果會越好，這裡因為不想要成是執行太久所以把 keyboard 的 iteration 設成 21。鴨子大概在第 7 次迭代後就看不太出來差別了，所以 iteration 設 7。

經過 RL 的 keyboard 像是法二和法三的中間值，比法三模糊一點，但是去除 motion 的程度比較好，也不會像法二那麼模糊。鴨子的部分就還是前面的方法效果比較好。
我認為我的 keyboard 在這個方法表現最好

Reference:

[1] Digital Image Processing International Edition, 4th Edition by R. C. Gonzalez.

[2] 老師上課的 ppt

[3] Discussion with 楊冠彥同學

[4] https://ieeexplore.ieee.org/document/8316595

[5] https://stackoverflow.com/questions/50474302/how-do-i-adjust-brightness-contrast-and-vibrance-with-opencv-python

[6] http://t.csdnimg.cn/epHqD

[7] https://github.com/opencv/opencv/blob/3.2.0/samples/python/deconvolution.py

[8] http://t.csdnimg.cn/ERPRH

[9] https://blog.csdn.net/youcans/article/details/123027356

[10] https://stackoverflow.com/questions/72023324/can-deconvolution-wiener-filter-reduce-noise-without-having-blurred-image-at-the

[11] https://blog.csdn.net/youcans/article/details/123062695

[12] https://dsp.stackexchange.com/questions/85102/convolving-image-with-kernel-with-fourier

[13] https://blog.csdn.net/youcans/article/details/123062903

[14] https://stackoverflow.com/questions/9854312/how-does-richardson-lucy-algorithm-work-code-example

[15] https://www.sciencedirect.com/science/article/pii/S2405844023045401