

目標：找出 X 光影像中骨骼的輪廓

輸入影像：896x1024

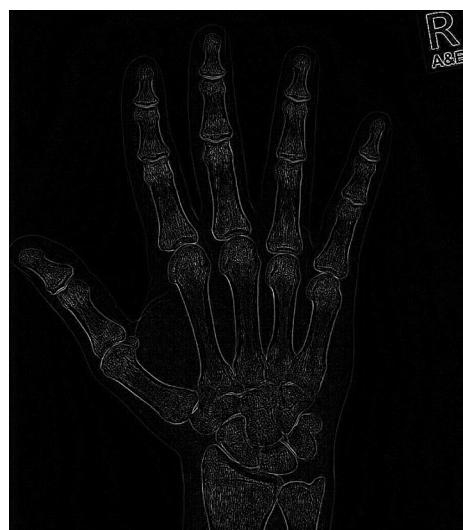
方法：Laplacian Filters

▼原圖



Laplacian Filters	
Kernel	
$\begin{array}{ c c c } \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline 1 & 1 & 1 \\ \hline 1 & -8 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$
Result	
稍微有一些白點但外型很不清楚	有輪廓但細節處不清楚

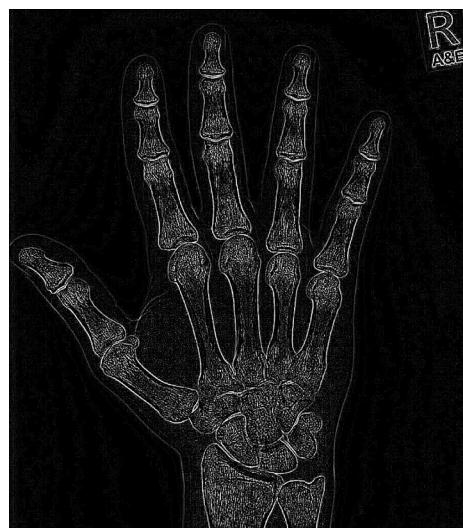
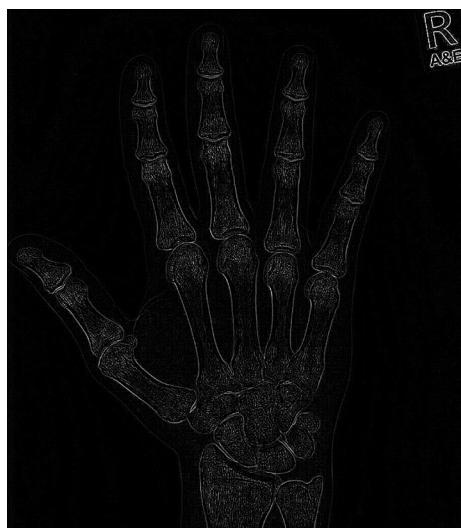
Result \* 2



有輪廓但細節處不清楚

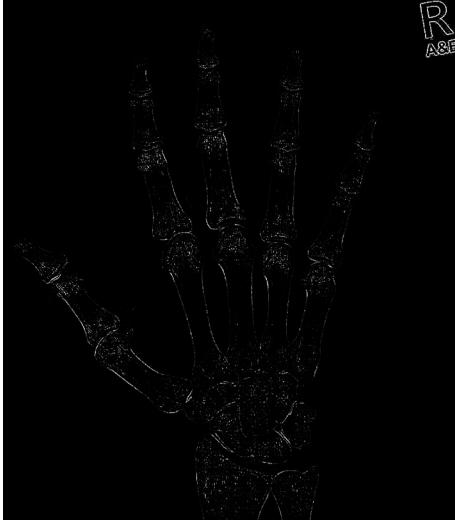
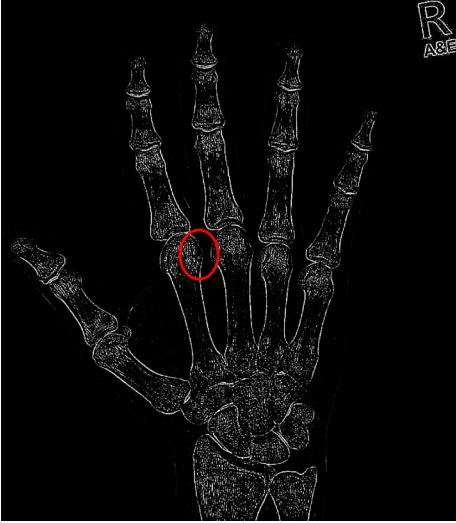
骨頭中的紋理也顯現出來了

Result \* 3



骨骼輪廓更清楚但內部紋理也更明顯

骨頭內部白點更多且非骨頭部分也顯現出來

1.2.^Result	1.1.^Result
	
1.3.^Result	1.15.^Result
	
部分輪廓有缺陷(斜向部分更嚴重)	輪廓更清楚，且非骨頭部分減少了 結論
使用這個 Kernel 斜向的輪廓容易顯現 不出來	斜向輪廓可以改善，跟 kernel 對角為 非 0 有關係

Kernel(其他嘗試)

$$\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 5 & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array}$$

$$\begin{array}{|c|c|c|}\hline -1 & -1 & -1 \\ \hline -1 & 9 & -1 \\ \hline -1 & -1 & -1 \\ \hline\end{array}$$

Result \* 2



看起來手部皮膚部分更明顯，骨骼紋理也更清晰

1.04 .^ Result



看起來像是移除皮膚部分並且加強骨骼的部分

使用 Matlab 實作

Code(Filter):

```
function imgResult = Filter(img, kernel)
    imgResult = img;
    sizeImg = size(img);
    sizeKernel = size(kernel);
    tmpImg = zeros(sizeImg+fix(sizeKernel/2)*2);
    tmpImg(fix(sizeKernel(1)/2)+1:fix(sizeKernel(1)/2)+sizeImg(1),
fix(sizeKernel(2)/2)+1:fix(sizeKernel(2)/2)+sizeImg(2)) = img;
    for row = 1:sizeImg(1)
        for col = 1:sizeImg(2)
            imgResult(row, col) = sum(tmpImg(row:row+sizeKernel(1)-1,
col:col+sizeKernel(2)-1) .* kernel, "all");
        end
    end
end
```

Code(X 光):

```
img= imread("xray.png");
img_gray = im2gray(img);

k = ones(3,3)*(-1);
k(1,1) = 0;
k(1,3) = 0;
k(3,1) = 0;
k(3,3) = 0;
k(2,2) = 5;
img_result = Filter(img_gray, k);

figure,imshow(img_gray,[]);
figure,imshow(1.04.^img_result,[]);

k = ones(3,3)*(-1);
k(2,2) = 9;
img_result = Filter(img_gray, k);

figure,imshow(1.04.^img_result,[]);
```

目標：去除影像中的噪點

方法：box filter(3x3)、averaging filter(3x3)、Gaussian Smoothing Filter(5x5)

原圖		
box filter		
原圖細節處		

box filter 細節處	 <p>天空的噪點變平滑但是城市的細節部分也變得不清楚了</p>	 <p>原圖中有的像素格再處理後就平滑化消失了</p>
averaging filter	 <p>與 box filter 差異不大</p>	 <p>與 box filter 差異不大，細節處的像素格也平滑消失了</p>
Gaussian Smoothing Filter		

使用 Matlab 實作

Code(去噪點)：

```
img= imread("camera.png");

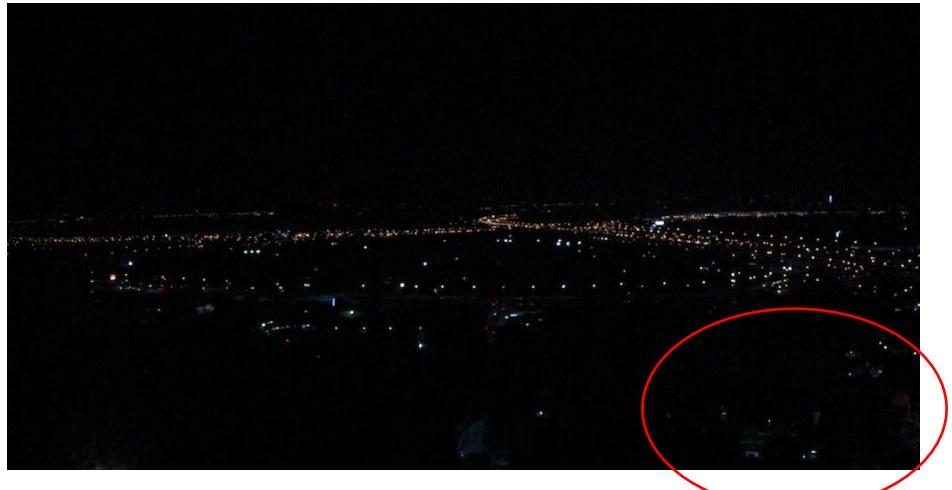
% Kernel(高斯平滑 5x5 kernel)
k = [1 4 7 4 1; 4 16 26 16 4; 7 26 41 26 7; 4 16 26 16 4; 1 4 7 4
1]./273;

img_result = img;
img_result(:,:1) = Filter(img(:,:1), k);
img_result(:,:2) = Filter(img(:,:2), k);
img_result(:,:3) = Filter(img(:,:3), k);

figure,imshow(img,[]);
figure,imshow(img_result,[]);
```

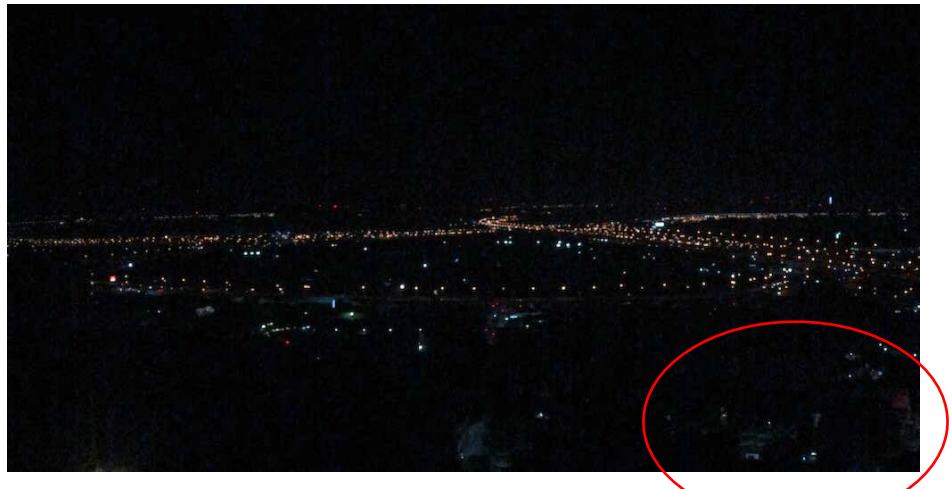
目標：改善夜景照片的視覺效果(看得更清楚)

▼原圖(只看的到燈光)

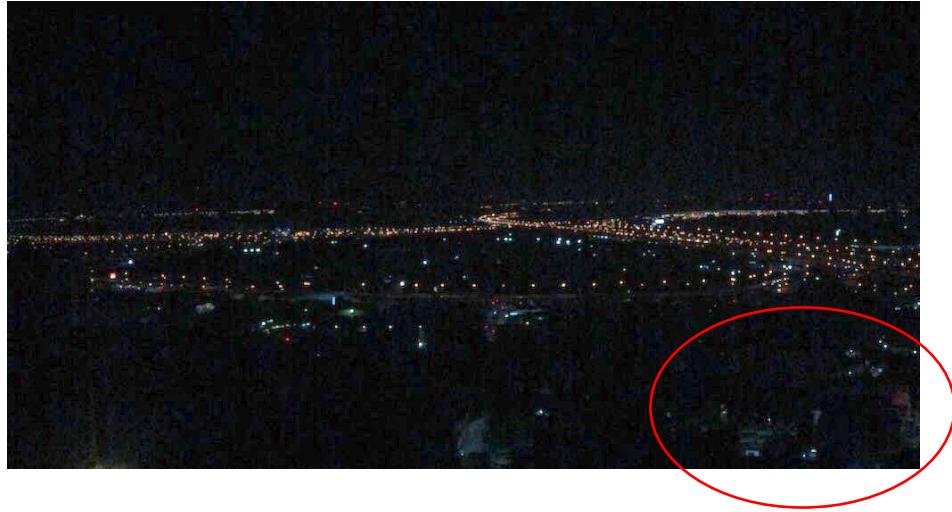


方法一、使用 Gamma correction 調整照片

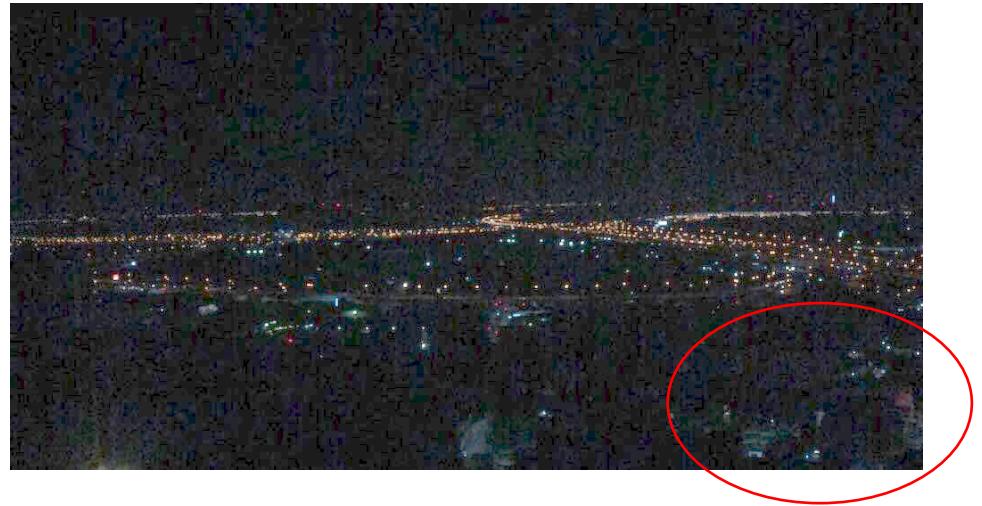
▼ $s=cr^{0.8}$



▼ $s=cr^{0.6}$



$$\nabla s = cr^{0.4}$$



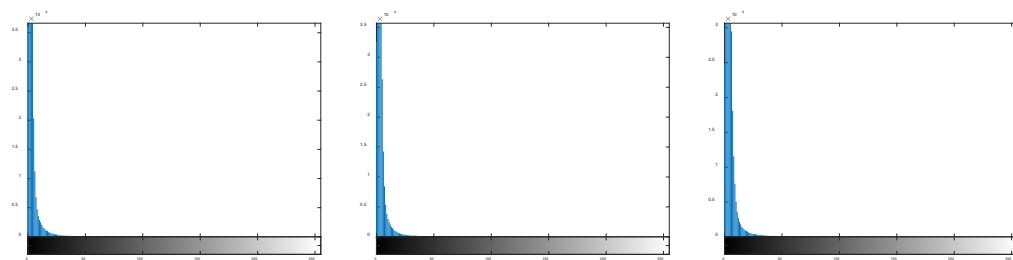
$$\nabla s = cr^{0.8} \sim s = cr^{0.4}$$



使用 Gamma correction 調整圖片，當  $\gamma$  越小暗的部分能夠顯示越多細節，但是小到一定程度後會開始出現噪點(像素間不連續，即  $s=cr^{0.4}$  時)，影像也會出現過曝的情形。

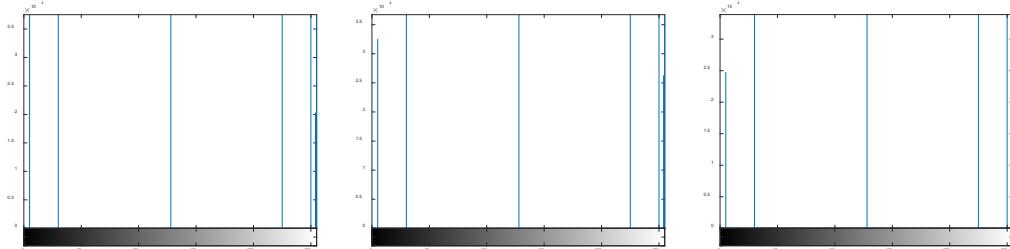
## 方法二、使用 Histogram Equalization 調整照片

$$\nabla \text{RGB 直方圖}$$

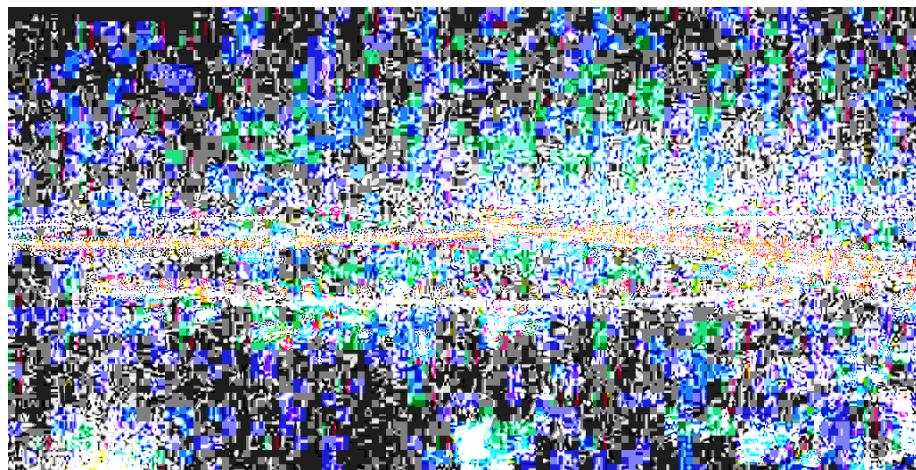


根據上面直方圖的分布，可以找一個 RGB 轉換函式： $y = \left( \frac{255}{1+e^{-2x+4}} \right)$ (根據 sigmoid function 去做變形)

▼轉換後直方圖



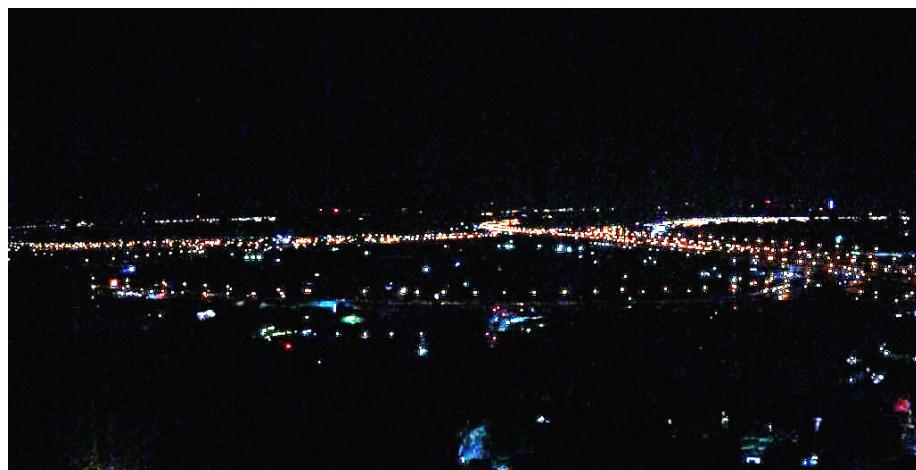
▼轉換後影像



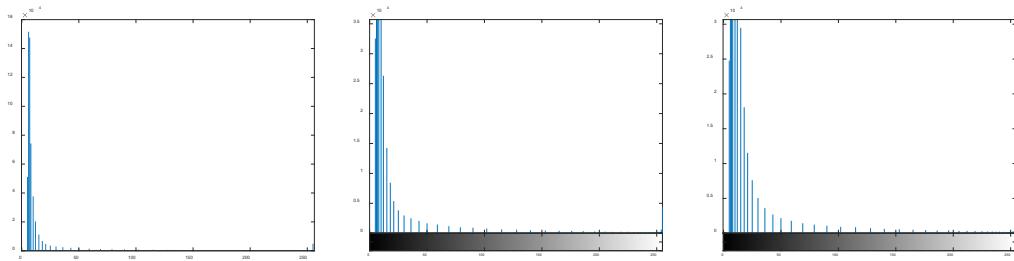
推測：RGB 值分布範圍大部分在 0-10，且 0-10 轉換後如下表，分布大部分為下列的值，導致影像會上圖容易產生較大的噪點，但是光點的輪廓能顯示出來。

x	0	1	2	3	4	5	6	7	8	9	10
y	5	30	128	224	250	254	255	255	255	255	255

- 修改 RGB 轉換函式為： $y = \left( \frac{255}{1+e^{-0.2x+4}} \right)$



▼轉換後直方圖

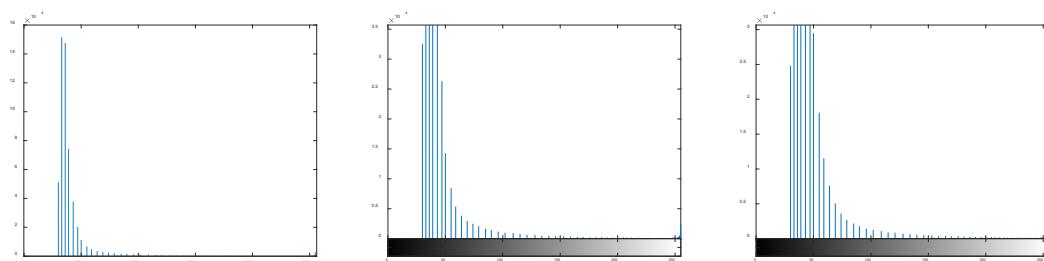


結果：值方圖分散的範圍更廣了，但是光點的地方顯得更擴大(亮與暗的對比變得更大了)

- 修改 RGB 轉換函式為： $y = \left(\frac{255}{1+e^{-0.1x+2}}\right)$



▼轉換後直方圖



結果：值方圖分散的範圍更廣了，且分布位置向右平移，導致影像變得更白。

使用 Matlab 實作

Code(方法一) :

```
img= imread("night_view.png");

img_f = double(img);
img_f = 255.0*(img_f./255.0).^0.6;
img_r = uint8(img_f);

figure,imshow(img,[]);
figure,imshow(img_r,[]);
```

Code(方法二) :

```
img= imread("night_view1.png");

img_f = double(img);
img_f = 255.0./(1+exp(1).^(-0.1.*img_f+2));
img_r = uint8(img_f);

figure,imshow(img);
figure,imshow(img_r);
figure,imHist(img_r(:,:,1));
figure,imhist(img_r(:,:,2));
figure,imhist(img_r(:,:,3));
```