R05631027_楊皓文 _影像處理 _作業 #4

編譯環境 : Windows 10 \ Matlab 2016

Part 1 :

程式碼 :

```
function varargout = HW01(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @HW01_OpeningFcn, ...
                   'gui_OutputFcn',  @HW01_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end



function HW01_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
guidata(hObject, handles);



function varargout = HW01_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;



function pushbutton1_Callback(hObject, eventdata, handles)

% 讀取圖片
filename = uigetfile({'*.jpg;*.tif;*.png;*.gif;*.bmp','All Image Files'});
handles.I = imread(filename);
```

```
%  轉換成灰階影像
[handles.row, handles.col, handles.lay] = size(handles.I);
if handles.lay == 1
        handles.I = handles.I;
        fprintf('1 layers image\n');
elseif handles.lay == 3
        handles.R = uint8(handles.I(:, :, 1));
        handles.G = uint8(handles.I(:, :, 2));
        handles.B = uint8(handles.I(:, :, 3));
        handles.I = 0.299*handles.R + 0.587*handles.G + 0.114*handles.B;
        fprintf('3 layers image\n');
end

axes(handles.axes1); imshow(handles.I);

%  顯示圖片大小
row = size(handles.I, 1);
col = size(handles.I, 2);
set(handles.text2, 'String', ['Image size : ', num2str( row )...
        ,' x ',num2str( col ) ] );

handles.output = hObject;
guidata(hObject, handles);



function pushbutton2_Callback(hObject, eventdata, handles)

% time start  計時開始
timer1=tic;

% fourier transform
handles.F = fft2(handles.I);

% time end  顯示時間
set(handles.text3, 'String', ['Overall time =', num2str( toc(timer1) ), ' sec'] );

%  使頻譜容易觀察
F_min = log( 1 + abs( min(min(handles.F)) ) );
F_max = log( 1 + abs( max(max(handles.F)) ) );
G = 255;
Y = G*( log(1+abs(handles.F)) - F_min )/( F_max - F_min );
Y = uint8(Y);
```

```
Y_shift = fftshift(Y);

axes(handles.axes2); imshow(Y_shift);

handles.output = hObject;
guidata(hObject, handles);



function pushbutton3_Callback(hObject, eventdata, handles)

% time start  計時開始
timer1=tic;

% inverse fourier transform
I2 = uint8( ifft2(handles.F) );

% time end  顯示時間
set(handles.text3, 'String', ['Overall time = ', num2str( toc(timer1) ), ' sec'] );

axes(handles.axes2); imshow(I2);

handles.output = hObject;
guidata(hObject, handles);
```

結果呈現與討論 ：

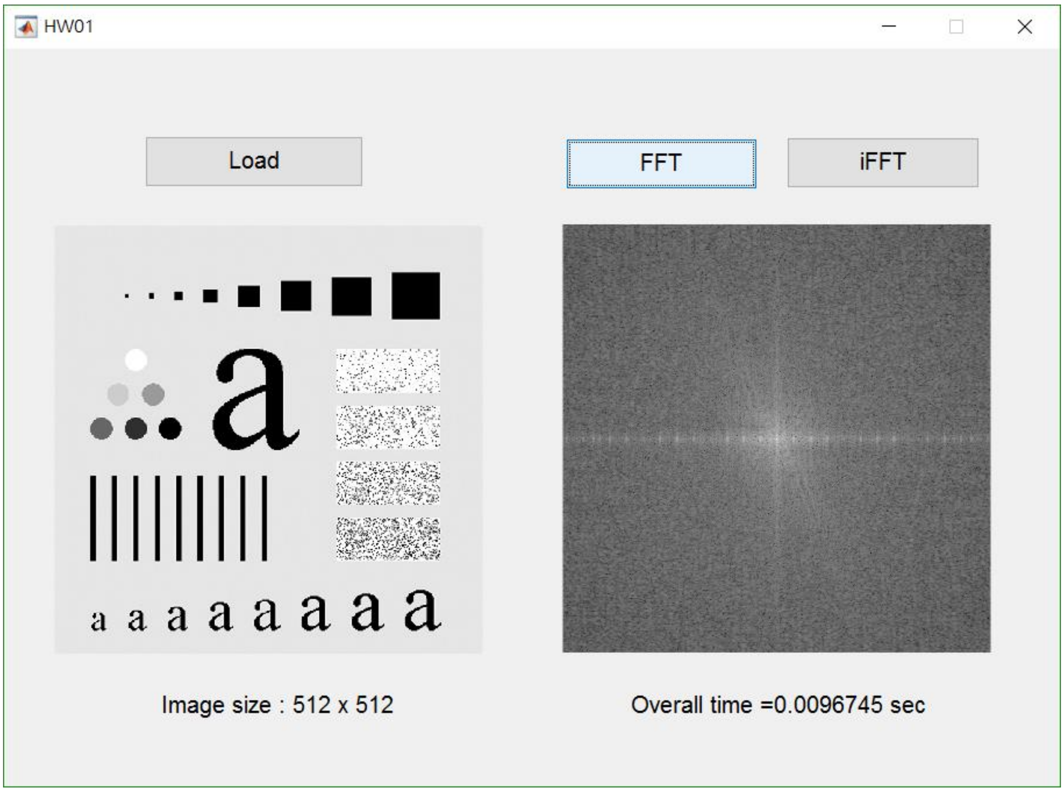


Image size : 512 x 512

Overall time =0.0096745 sec

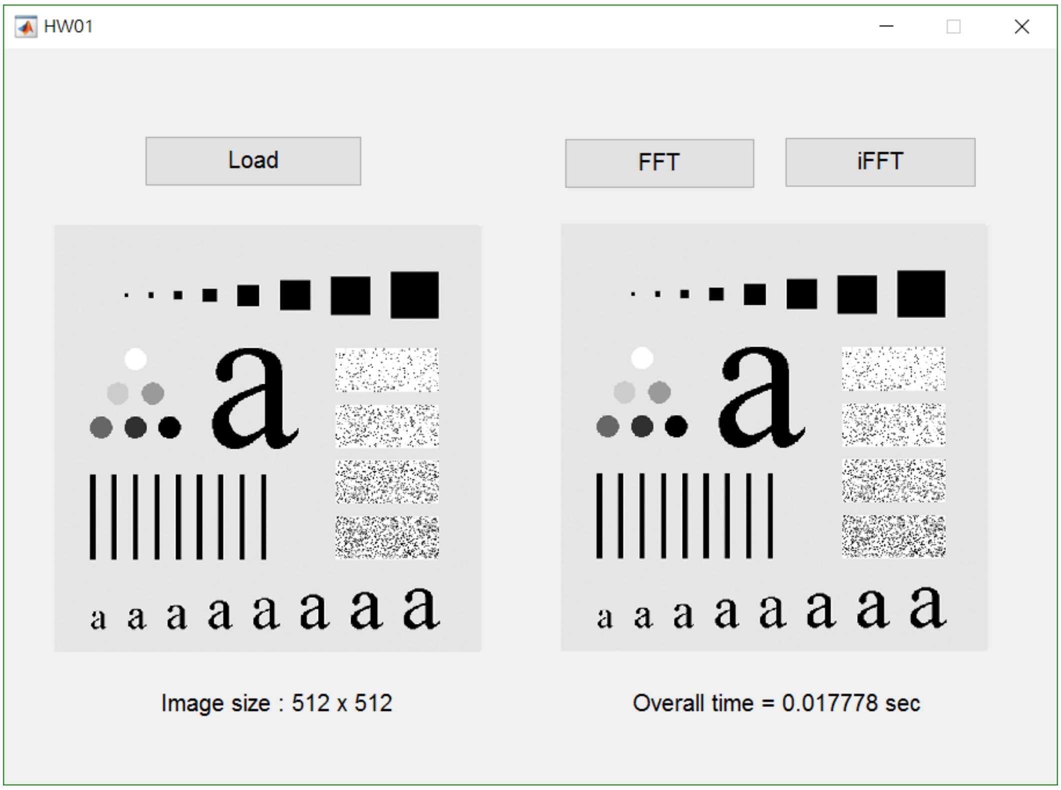


Image size : 512 x 512

Overall time = 0.017778 sec

由上面兩張結果圖可以看到原始影像順利轉換成頻譜影像，且經過傅立葉反轉換後，圖片不會改變，會變回原始影像。

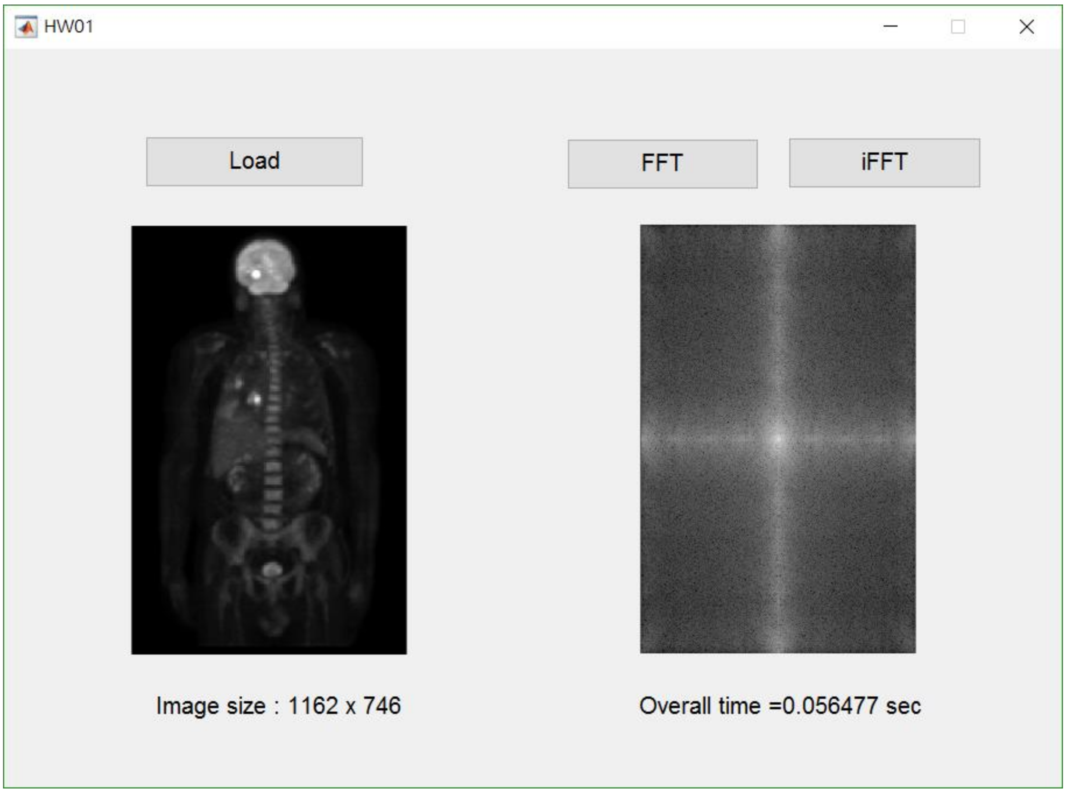


Image size : 1162 x 746

Overall time =0.056477 sec

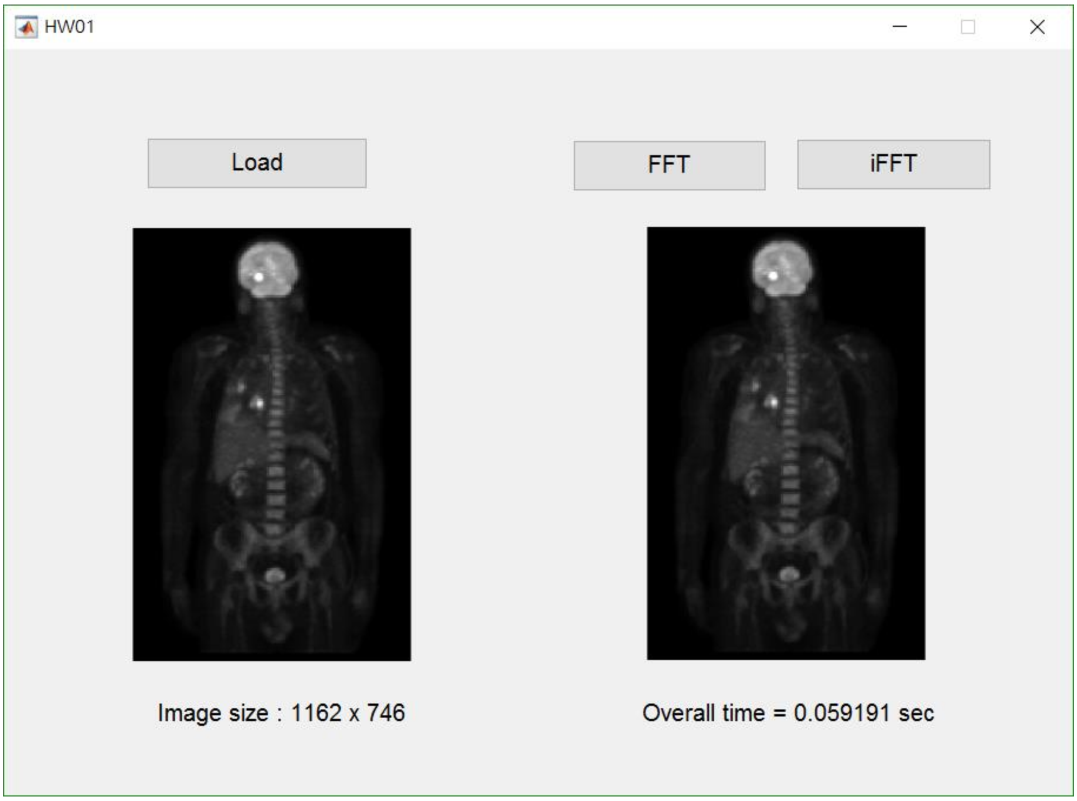


Image size : 1162 x 746

Overall time = 0.059191 sec

另一方面，可以發現傅立葉正逆轉換所花費的時間差不多，但使用較大的影像後，轉換時間則明顯變長。

Part 2 :

程式碼 ：

```
function varargout = HW02(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',          mfilename, ...
                   'gui_Singleton',   gui_Singleton, ...
                   'gui_OpeningFcn', @HW02_OpeningFcn, ...
                   'gui_OutputFcn',   @HW02_OutputFcn, ...
                   'gui_LayoutFcn',   [] , ...
                   'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
function HW02_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
guidata(hObject, handles);



function varargout = HW02_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;



function pushbutton1_Callback(hObject, eventdata, handles)

% 讀取圖片
filename = uigetfile({'*.jpg;*.tif;*.png;*.gif;*.bmp','All Image Files'});
handles.I = imread(filename);

% 轉成灰階
[handles.row, handles.col, handles.lay] = size(handles.I);

if handles.lay == 1
    handles.I = handles.I;
    fprintf('1 layers image\n');
elseif handles.lay == 3
    handles.R = uint8(handles.I(:, :, 1));
    handles.G = uint8(handles.I(:, :, 2));
    handles.B = uint8(handles.I(:, :, 3));
    handles.I = 0.299*handles.R + 0.587*handles.G + 0.114*handles.B;
    fprintf('3 layers image\n');
end

axes(handles.axes1); imshow(handles.I);

handles.output = hObject;
guidata(hObject, handles);



function radiobutton1_Callback(hObject, eventdata, handles)

set(handles.radiobutton1, 'Value', 1);
set(handles.radiobutton2, 'Value', 0);
```

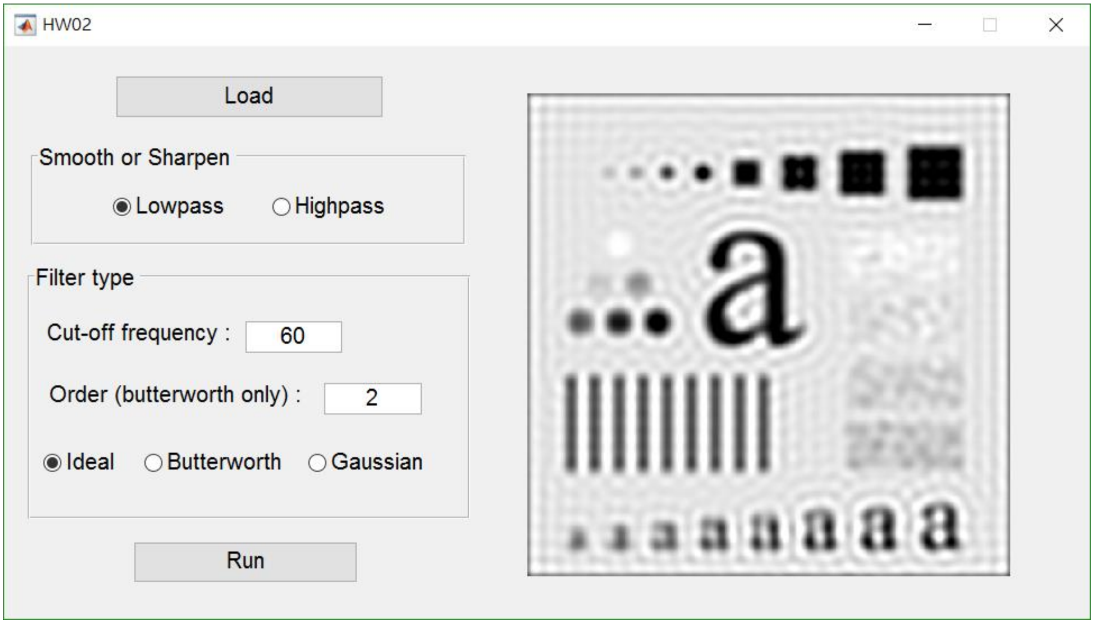
```
function radiobutton2_Callback(hObject, eventdata, handles)

set(handles.radiobutton1, 'Value', 0);
set(handles.radiobutton2, 'Value', 1);


function radiobutton3_Callback(hObject, eventdata, handles)

set(handles.radiobutton3, 'Value', 1);
set(handles.radiobutton4, 'Value', 0);
set(handles.radiobutton5, 'Value', 0);


% --- Executes on button press in radiobutton4.
function radiobutton4_Callback(hObject, eventdata, handles)

set(handles.radiobutton3, 'Value', 0);
set(handles.radiobutton4, 'Value', 1);
set(handles.radiobutton5, 'Value', 0);


function radiobutton5_Callback(hObject, eventdata, handles)

set(handles.radiobutton3, 'Value', 0);
set(handles.radiobutton4, 'Value', 0);
set(handles.radiobutton5, 'Value', 1);


function edit1_Callback(hObject, eventdata, handles)


function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
     set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
function edit2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)


function edit3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)


function edit4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
end


function filter_H(hObject, eventdata, handles)

% cut-off frequency (D0), order (n)
D0 = str2double( get(handles.edit2, 'String') );
n = str2double( get(handles.edit3, 'String') );

[M, N] = size(handles.I);
P = 2*M;
Q = 2*N;

% filter
handles.H = zeros(P, Q);
```

```
D = zeros(P, Q);

% Lowpass
% ideal
if handles.radiobutton1.Value == 1 && handles.radiobutton3.Value == 1
    for i = 1:P
        for j = 1:Q
            D(i, j) = ( (i - P/2)^2 + ( j - Q/2 )^2 )^(1/2);
            if D(i, j) <= D0
                handles.H(i, j) = 1;
            end
        end
    end

% butterworth
elseif handles.radiobutton1.Value == 1 && handles.radiobutton4.Value == 1
    for i = 1:P
        for j = 1:Q
            D(i, j) = ( (i - P/2)^2 + ( j - Q/2 )^2 )^(1/2);
            handles.H(i, j) = 1 / ( 1 + ( D(i, j)/D0 )^(2*n) );
        end
    end

% gaussian
elseif handles.radiobutton1.Value == 1 && handles.radiobutton5.Value == 1
    for i = 1:P
        for j = 1:Q
            D(i, j) = ( (i - P/2)^2 + ( j - Q/2 )^2 )^(1/2);
            handles.H(i, j) = exp( -( D(i, j) )^2 / ( 2*D0^2 ) );
        end
    end

% ideal
elseif handles.radiobutton2.Value == 1 && handles.radiobutton3.Value == 1
    for i = 1:P
        for j = 1:Q
            D(i, j) = ( (i - P/2)^2 + ( j - Q/2 )^2 )^(1/2);
            if D(i, j) > D0
                handles.H(i, j) = 1;
            end
        end
    end
```

```
% butterworth
elseif handles.radiobutton2.Value == 1 && handles.radiobutton4.Value == 1
        for i = 1:P
                for j = 1:Q
                        D(i, j) = ( (i - P/2)^2 + ( j - Q/2 )^2 )^(1/2);
                        handles.H(i, j) = 1 / ( 1 + ( D0/D(i, j) )^(2*n) );
                end
        end

% gaussian
elseif handles.radiobutton2.Value == 1 && handles.radiobutton5.Value == 1
        for i = 1:P
                for j = 1:Q
                        D(i, j) = ( (i - P/2)^2 + ( j - Q/2 )^2 )^(1/2);
                        handles.H(i, j) = 1 - exp( -( D(i, j) )^2 / ( 2*D0^2 ) );
                end
        end


end

handles.output = hObject;
guidata(hObject, handles);


function pushbutton2_Callback(hObject, eventdata, handles)

% image processing
% step1
[M, N] = size(handles.I);
P = 2*M;
Q = 2*N;

% step2
fp = zeros(P, Q);
for i = 1:M
        for j = 1:N
                fp(i, j) = handles.I(i, j);
        end
end
```

```
% step3
for i = 1:M
        for j = 1:N
                fp(i, j) = fp(i, j)*( (-1)^(i+j) );
        end
end

% step4
F = fft2( fp );

% step5
filter_H(hObject, eventdata, handles);
handles = guidata(hObject);
G = handles.H .* F;

% step6
gp = zeros(P, Q);

Gifft = real( ifft2(G) );
for i = 1:P
        for j = 1:Q
                gp(i, j) = ( Gifft(i, j) )*(-1)^(i+j);
        end
end

% step7
g = zeros(M, N);
for i = 1:M
        for j = 1:N
                g(i, j) = gp(i, j);
        end
end
axes(handles.axes1); imshow( uint8(g) );

handles.output = hObject;
guidata(hObject, handles);
```

結果呈現與討論 ：



Smooth or Sharpen

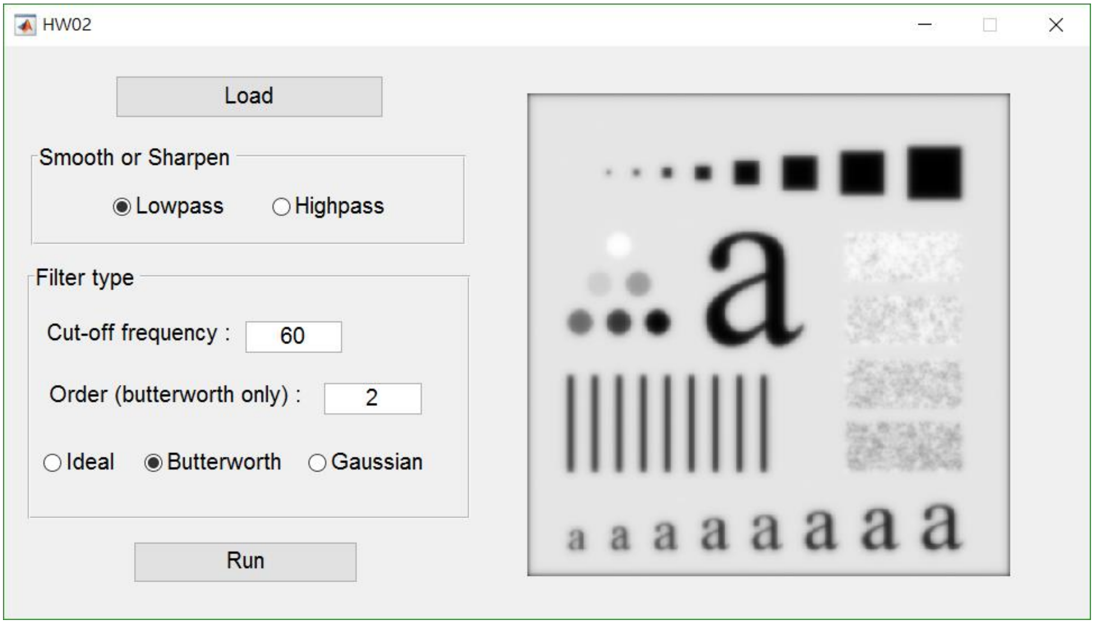
● Lowpass ○ Highpass

Filter type

Cut-off frequency : 60

Order (butterworth only) : 2

● Ideal ○ Butterworth ○ Gaussian



Smooth or Sharpen

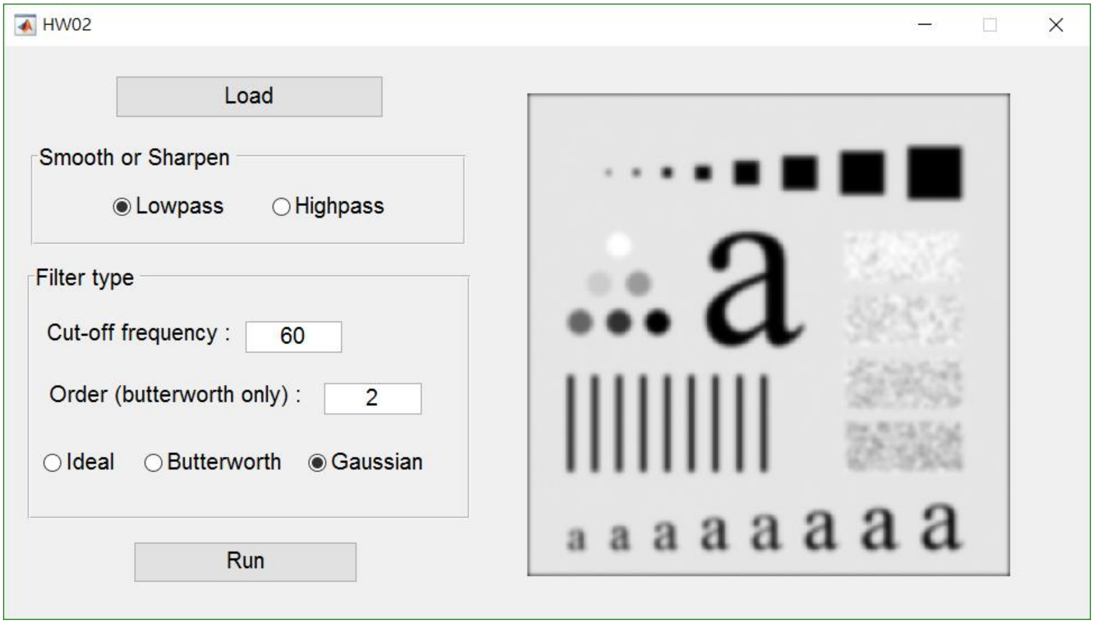
● Lowpass ○ Highpass

Filter type

Cut-off frequency : 60

Order (butterworth only) : 2

○ Ideal ● Butterworth ○ Gaussian



Smooth or Sharpen

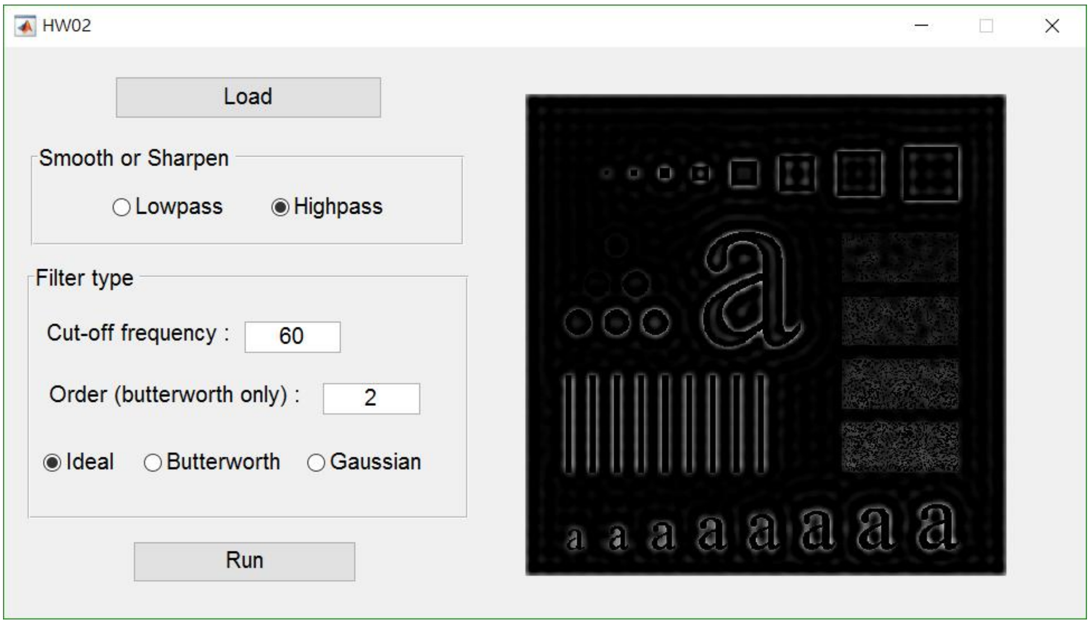
● Lowpass ○ Highpass

Filter type

Cut-off frequency : 60

Order (butterworth only) : 2

○ Ideal ○ Butterworth ● Gaussian



Smooth or Sharpen

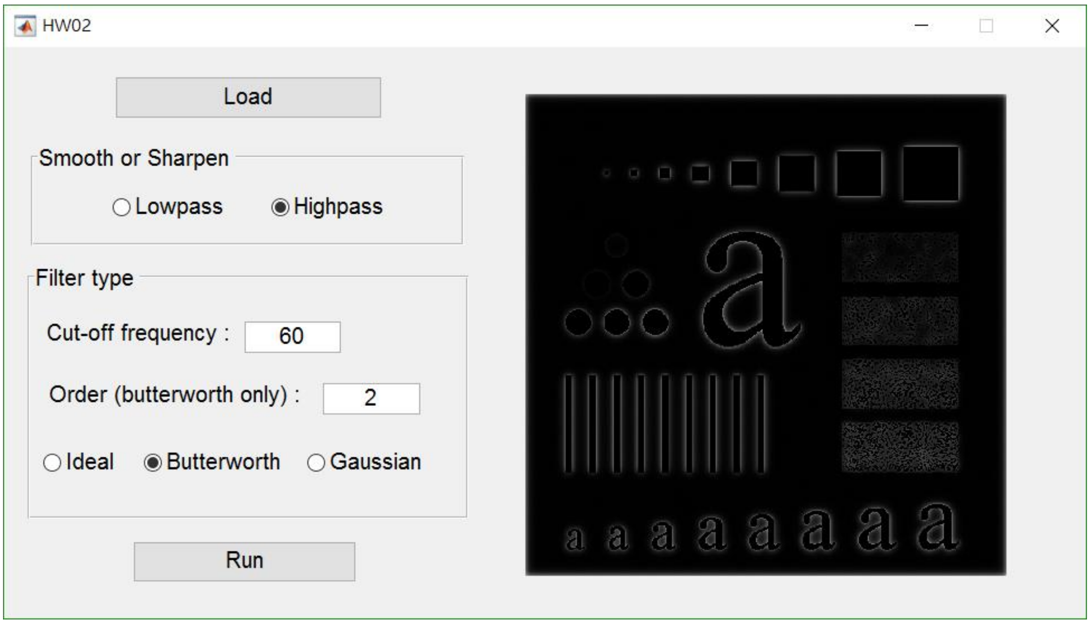
○ Lowpass ● Highpass

Filter type

Cut-off frequency : 60

Order (butterworth only) : 2

● Ideal ○ Butterworth ○ Gaussian



Smooth or Sharpen

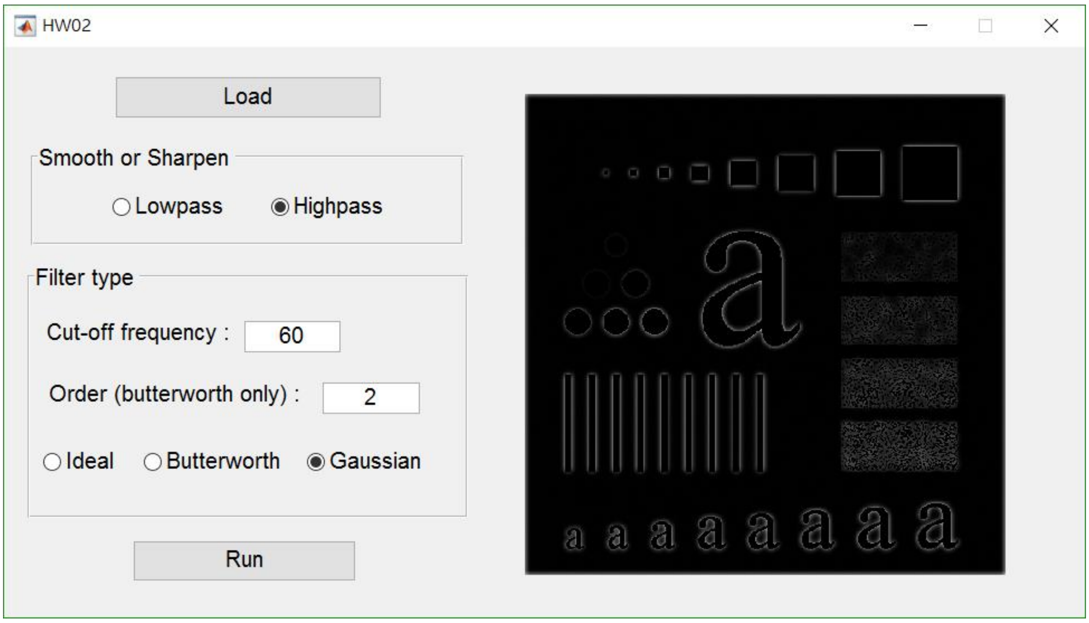
○ Lowpass ● Highpass

Filter type

Cut-off frequency : 60

Order (butterworth only) : 2

○ Ideal ● Butterworth ○ Gaussian



Smooth or Sharpen

○ Lowpass ● Highpass

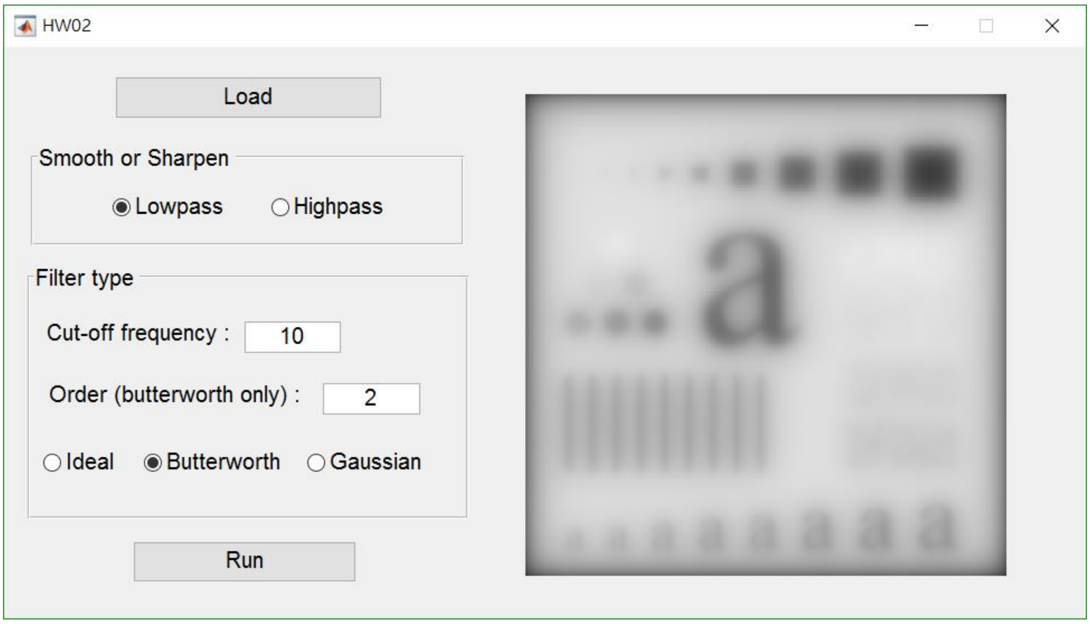
Filter type

Cut-off frequency : 60

Order (butterworth only) : 2

○ Ideal ○ Butterworth ● Gaussian

上面六張圖為高通低通和三種濾波器的排列組合產生的結果，在相同的 cut-off frequency 下，ideal filter 的影像較為多雜訊，而 butterworth 和 gaussain filter 則有點類似，模糊的效果比較平滑。



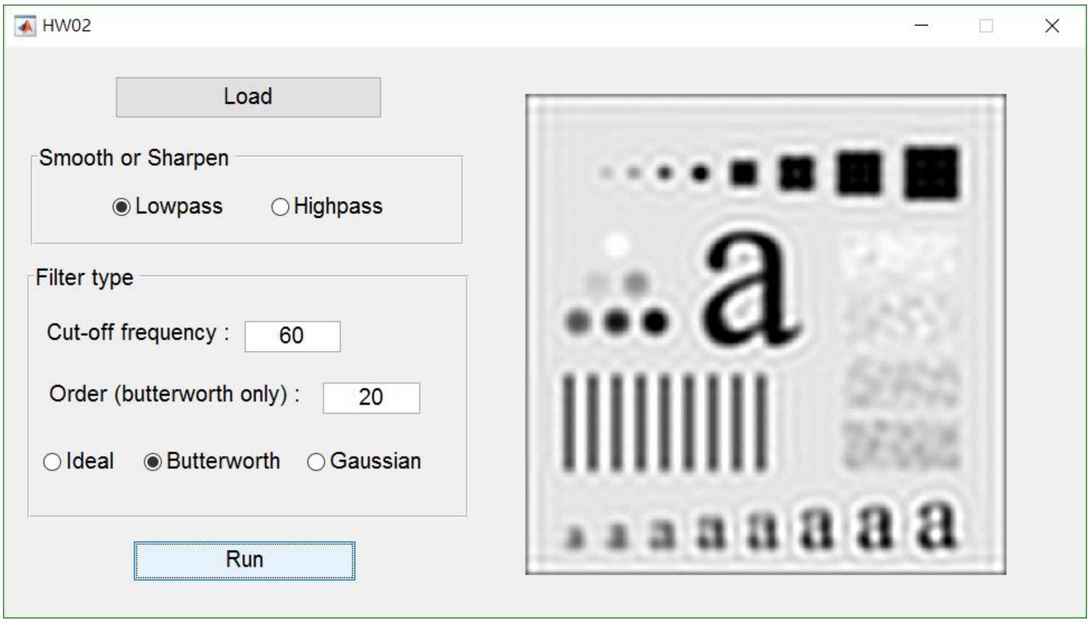
● Lowpass ○ Highpass

Cut-off frequency : 10

Order (butterworth only) : 2

○ Ideal ● Butterworth ○ Gaussian

以 butterworth lowpass 當例子，當 cut-off frequency 由 60 變成 10，order 不變，我們可以發現影像明顯的便更模糊了。



● Lowpass ○ Highpass

Cut-off frequency : 60

Order (butterworth only) : 20

○ Ideal ● Butterworth ○ Gaussian

以 butterworth lowpass 當例子，當 order 由 2 變 20，cut-off frequency 不變，我們可以看到影像有種光暈的效果，有點像散光的感覺。

Part 3 :

程式碼 ：

```
function varargout = HW03(varargin)


gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @HW03_OpeningFcn, ...
                   'gui_OutputFcn',  @HW03_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
```

```
if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
        gui_mainfcn(gui_State, varargin{:});
end



function HW03_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;
guidata(hObject, handles);



function varargout = HW03_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;



function pushbutton1_Callback(hObject, eventdata, handles)

%  讀取檔案
filename = uigetfile({'*.jpg;*.tif;*.png;*.gif;*.bmp','All Image Files'});
handles.I = imread(filename);

%  轉成灰階
[handles.row, handles.col, handles.lay] = size(handles.I);

if handles.lay == 1
        handles.I = handles.I;
        fprintf('1 layers image\n');
elseif handles.lay == 3
        handles.R = uint8(handles.I(:, :, 1));
        handles.G = uint8(handles.I(:, :, 2));
        handles.B = uint8(handles.I(:, :, 3));
        handles.I = 0.299*handles.R + 0.587*handles.G + 0.114*handles.B;
        fprintf('3 layers image\n');
end
```

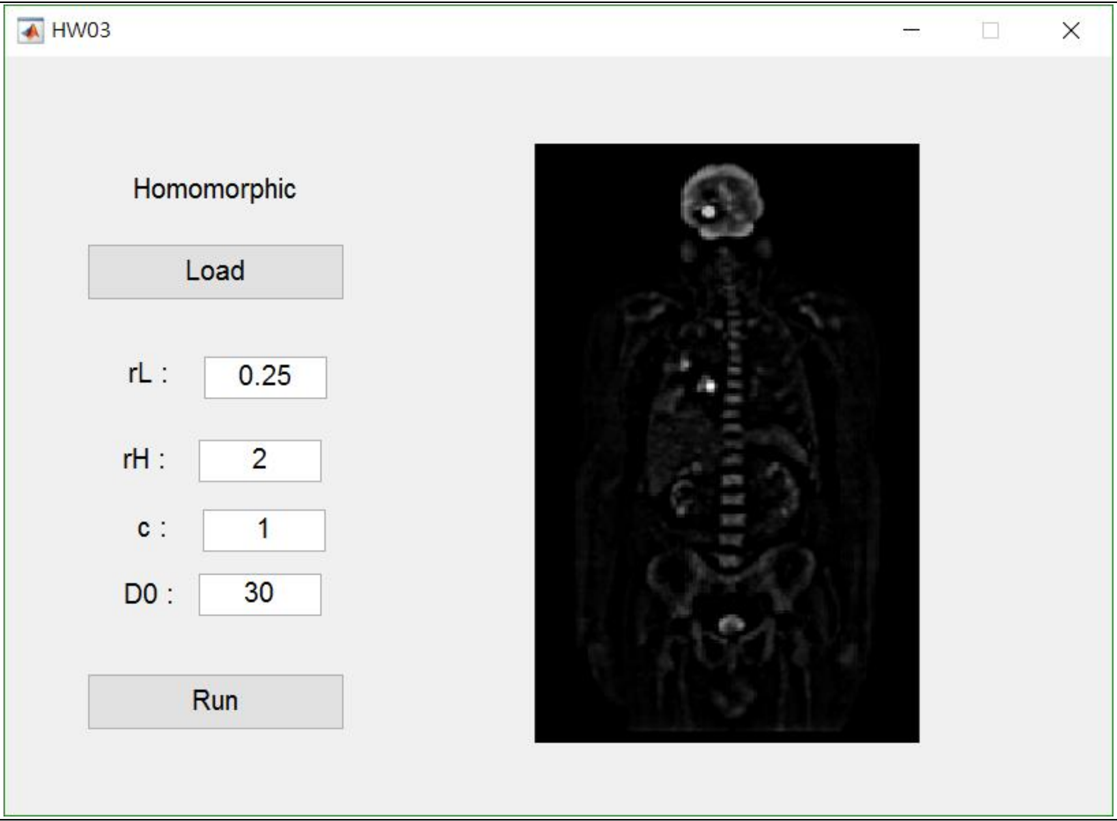
```
axes(handles.axes1); imshow(handles.I);

handles.output = hObject;
guidata(hObject, handles);



% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)

% image processing
rL = str2double( get(handles.edit1, 'String') );
rH = str2double( get(handles.edit2, 'String') );
c = str2double( get(handles.edit3, 'String') );
D0 = str2double( get(handles.edit4, 'String') );

% step1
[M, N] = size(handles.I);
P = 2*M;
Q = 2*N;

% step2
fp = zeros(P, Q);
for i = 1:M
    for j = 1:N
        fp(i, j) = handles.I(i, j);
    end
end

% step3
for i = 1:M
    for j = 1:N
        fp(i, j) = fp(i, j)*( (-1)^(i+j) );
    end
end

% step4
F = fft2( fp );

% step5
D = zeros(P, Q);
H = zeros(P, Q);
% Homomorphic filter
```

```
for i = 1:P
    for j = 1:Q
        D(i, j) = ( (i - P/2)^2 + ( j - Q/2 )^2 )^(1/2);
        H(i, j) = ( rH - rL )*(1 - exp( -c*( (D(i, j))^2 / D0^2 ) )) + rL;
    end
end

G = H .* F;

% step6
gp = zeros(P, Q);

Gifft = real( ifft2(G) );
for i = 1:P
    for j = 1:Q
        gp(i, j) = ( Gifft(i, j) )*(-1)^(i+j);
    end
end

% step7
g = zeros(M, N);
for i = 1:M
    for j = 1:N
        g(i, j) = gp(i, j);
    end
end

axes(handles.axes1); imshow( uint8(g) );

handles.output = hObject;
guidata(hObject, handles);



function edit1_Callback(hObject, eventdata, handles)



function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit2_Callback(hObject, eventdata, handles)


function edit2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit3_Callback(hObject, eventdata, handles)


function edit3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function edit4_Callback(hObject, eventdata, handles)


function edit4_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

結果呈現 ：

將 rL 由 0.25 提高到 1，可以發現影像的整體亮度提高，身體骨骼的部分可以看得更清楚。



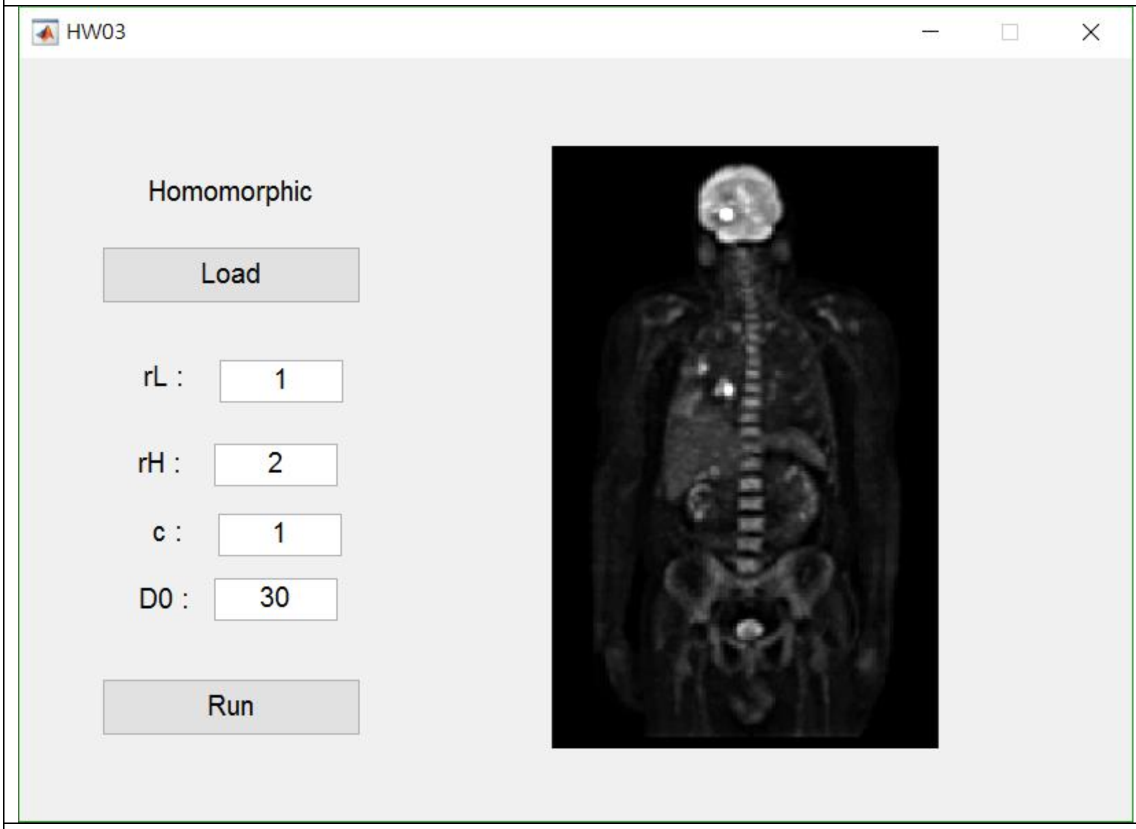
rL : 0.25

rH : 2

c : 1

D0 : 30

經過 homomorphic filter 後，我們可以發現影像的邊緣輪廓更明顯，像是頭腦的部分，可以更明顯的看出有個小圓形的亮點。



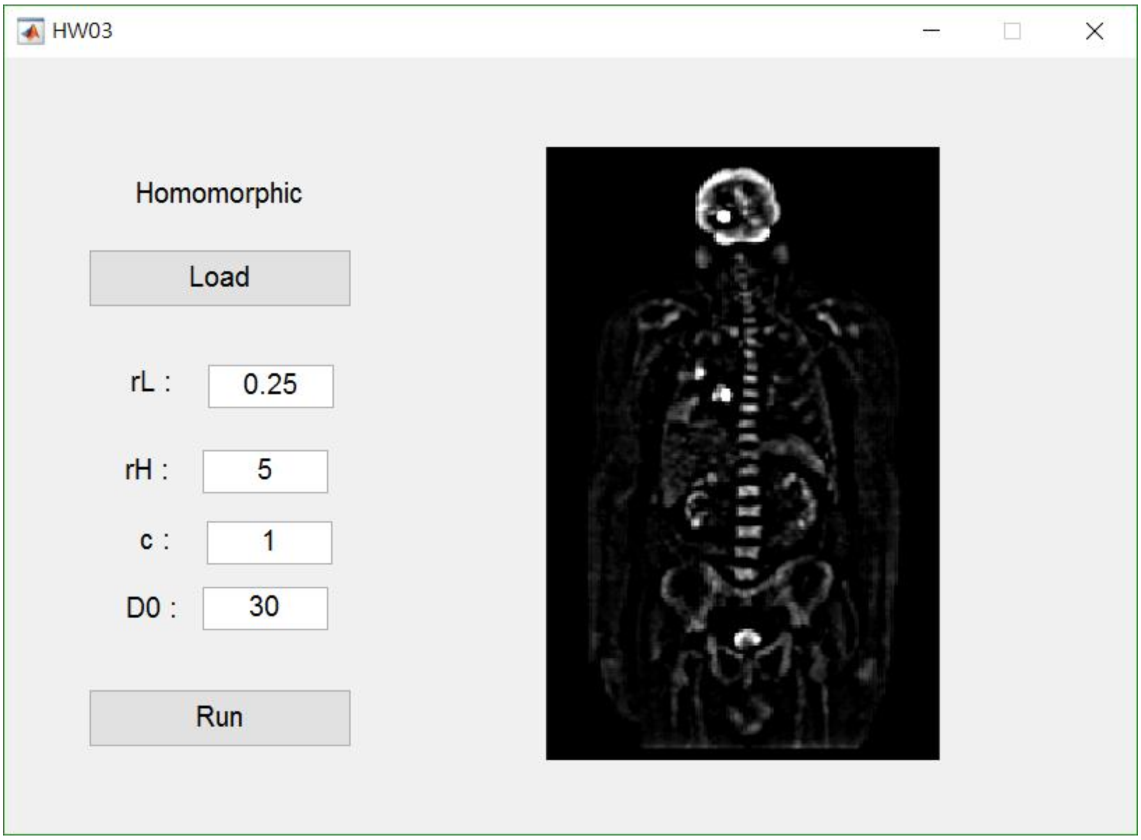
rL : 1

rH : 2

c : 1

D0 : 30

將 rL 由 0.25 提高到 1，可以發現影像的整體亮度提高，身體骨骼的部分可以看得更清楚。



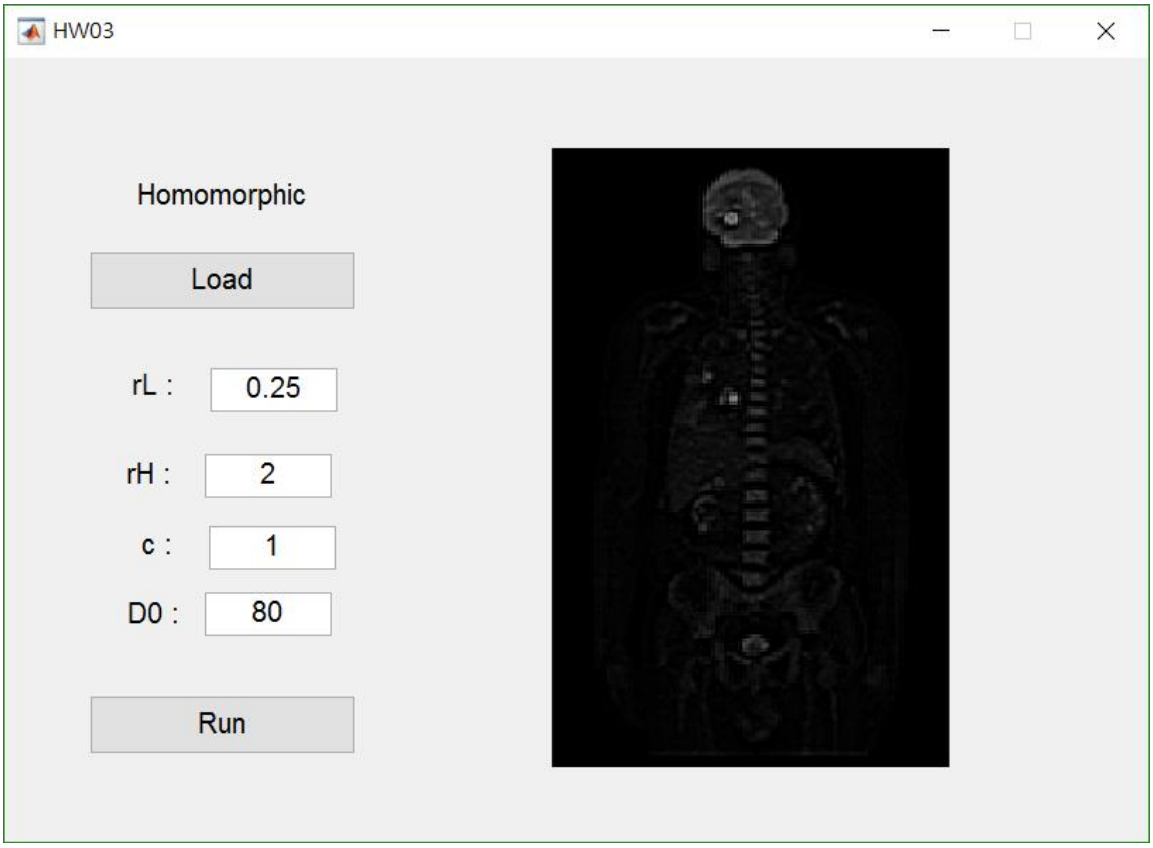
rL : 0.25

rH : 5

c : 1

D0 : 30

將 rH 由 2 提高到 5，可以發現影像的輪廓線更深，也更粗，對比更加明顯。



rL : 0.25

rH : 2

c : 1

D0 : 80

將 D0 由 30 提高到 80，可以看出影像有點過暗，對比也不是很清楚。