```
clear all
*********************************************************************
clc

%% reading the image
Princess = imread('angelina100.jpg');
%%imshow(Princess)


%% Going to GrayScale to Reduce Computations

GrPrincess = rgb2gray(Princess);
imshow(GrPrincess)


%% Applying the Mean Filter.

GrPrincess = MeanFilter(GrPrincess,15,1);
figure
imshow(GrPrincess);

%% Applying the Convolution Operation

GPrincess = ConvolveWithMask(GrPrincess);
%% Normalizing the Values
GPrincess = ((GPrincess/(max(max(GPrincess))-min(min(GPrincess))))*100);
GPrincess = uint8(GPrincess);

GPrincess = GPrincess*15;
%% Thresholding
%%Thresholds left as they were leading to very thick edges

%GPrincess = Threshold(GPrincess,20);
%GPrincess = GPrincess*255;
%% Output
GrPrincess = GrPrincess + GPrincess;
imshow(GPrincess);



*********************************************************************
```

ABOVE: The Main File

```
function IMAGE = MeanFilter(Im,MaskSize,weight)
*******************************************************************

IMAGE=Im;

Mask(MaskSize,MaskSize)=0;
Mask(:,:)=weight/(MaskSize*MaskSize);

for row = (ceil(MaskSize/2)):1:size(Im,1)-ceil(MaskSize/2)
    for col = (ceil(MaskSize/2)):1:size(Im,2)-ceil(MaskSize/2)

        FromImage = double(Im((row-
floor(MaskSize/2)):(row+floor(MaskSize/2)),(col-
floor(MaskSize/2)):(col+floor(MaskSize/2))));

        AValue = Mask.*FromImage;
        IMAGE(row,col) = sum(sum(AValue,2),1);
    end
end

*******************************************************************
```

ABOVE: Mean Filter Function Code

```
*******************************************************************
function IMAGE = ConvolveWithMask(Im)
TempIm = double(Im);
GradMag=Im;
MaskX = [-1 1];
MaskY = [-1;1];

%Using the Differential Model

for row = 2:1:size(Im,1)
    for col = 2:1:size(Im,2)
      EvalGradMag = sqrt((sum((TempIm(row-1:row,col).*MaskY)).^2)+
(sum((TempIm(row,col-1:col).*MaskX)).^2));
      IMAGE(row,col) = EvalGradMag;
    end
end

*******************************************************************
```

ABOVE: Convolving Mask for Differential

```
*********************************************************************

function IMAGE = Threshold(Im,Level)

IMAGE = Im;
Value = (max(max(Im))/100)*Level;

for row = 1:size(Im,1)
    for col = 1:size(Im,2)
        if Im(row,col)>=Value
            IMAGE(row,col)=1;
        else
            IMAGE(row,col)=0;
        end
    end
end
*********************************************************************
```

ABOVE: A Threshold based Reduction Function