```
%Pratap Luitel
%ENGS 111
%HW 4, Problem 1

%This script erodes or dilates the input image.
%It is assumed that the input image is of type double.
%Erode - flag 0, dilate - flag 1;
function imOut = erodeDilate(imIn,SE,flag)

[nRow,nCol]=size(imIn);

%find the center of the matrix
padLengthX = floor((size(SE,1)/2));
padLengthY = floor((size(SE,2)/2));

%padding
if (flag == 0)
    %pad with maximum value of the data type for erosion
    tempIm = padarray(imIn,[padLengthX padLengthY],1);
else
    %pad with minimum value of the data type for dilation
    tempIm = padarray(imIn,[padLengthX padLengthY],0);
end

imOut = zeros(size(imIn));
for i = 1:nRow
    for j= 1:nCol
        %imout of the same size as SE
        matrixToCheck = tempIm(i:i+(2*padLengthX),j:j+(2*padLengthY));
        index = find(SE == 1);
        if flag == 0
            imOut(i,j) = min(min(matrixToCheck(index)));  %#ok<*FNDSB>
        else
            imOut(i,j) = max(max(matrixToCheck(index)));
        end

    end
end


end
```

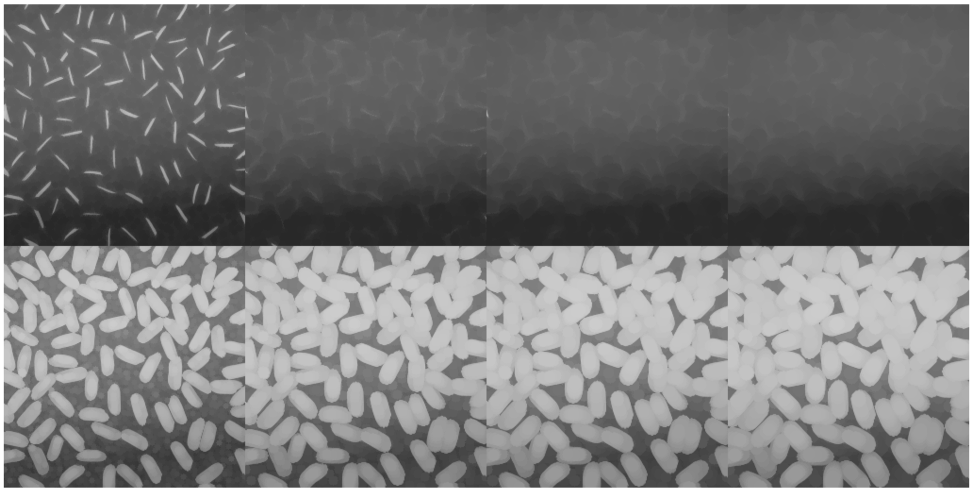
```
        Error using erodeDilate (line 10)
        Not enough input arguments.
```

Published with MATLAB® R2014a

```
%Pratap Luitel
%ENGS 111
%HW 4


im1 = imread('rice.png');
[i2,m2] = imread('son1-1.gif');

%convert to type double
imIn1 = im2double(im1);
imIn2 = ind2gray(i2,m2);
imIn2 = im2double(imIn2);

imageSequence1 = zeros(size(imIn1,1),size(imIn1,2),1,8);
imageSequence2 = zeros(size(imIn2,1),size(imIn2,2),1,8);

for i = 1:4
    n = [7 11 13 15];

    startX = floor(n(i)/2);
    x = repmat(-startX:startX,[n(i) 1]);
    y = repmat([-startX:startX]',[1 n(i)]);
    r = sqrt(x.^2 + y.^2);

    SE = zeros(n(i),n(i));
    SE(r <= startX) = 1;

    %rice.png
    imageSequence1(:,:,1,i) = erodeDilate(imIn1,SE,0); %erode
    imageSequence1(:,:,1,i+4) = erodeDilate(imIn1,SE,1); %dilate

    %son1-1.gif
    imageSequence2(:,:,1,i) = erodeDilate(imIn2,SE,0); %erode
    imageSequence2(:,:,1,i+4) = erodeDilate(imIn2,SE,1); %dilate

end
figure(1)
montage(imageSequence1,'size',[2,4]);
figure(2)
montage(imageSequence2,'size',[2 4]);
```

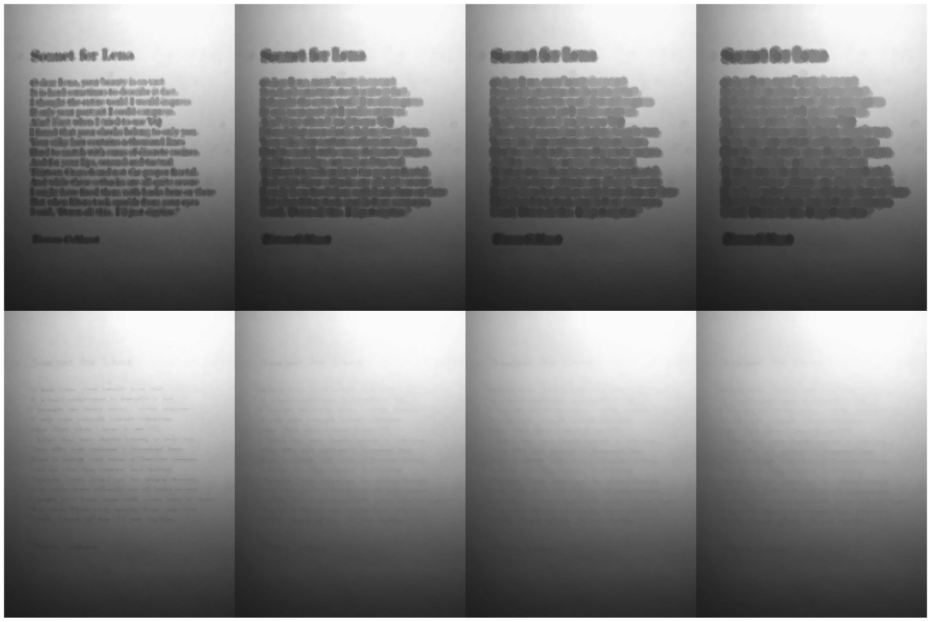
```
        Warning: Image is too big to fit on screen; displaying at 50%
```

Erode, size(SE) : 7 x 7 9 x 9 11 x 11 15x 15



Dilate, size(SE): 7 x 7 9 x 9 11 x 11 15 x 15

Erode, size(SE) : 7 x 7 9 x 9 11 x 11 15x 15



Dilate, size(SE) : 7 x 7 9 x 9 11 x 11 15x 15

Published with MATLAB® R2014a

```
%Pratap Luitel
%ENGS 111
%HW 4, Problem 2


%The function returns a filtered(top hat/bottom hat) image from
%the input image. The flag argument determines the choice of filter - 0
%is bottom hat and 1 is top hat. The function assumes that the input image
%is of type double.

function imOut = topBottomHat(imIn,SE,flag)

%bottom hat filtering
if (flag == 0)
    %closing = dilation, and then erosion
    dilatedImage = erodeDilate(imIn,SE,1);%dilate
    closeImage = erodeDilate(dilatedImage,SE,0);%erode
    imOut = imsubtract(closeImage,imIn);%bottomHat

%top hat filtering
else
    %opening = erosion and then dilation
    erodedImage = erodeDilate(imIn,SE,0);%erode
    openImage = erodeDilate(erodedImage,SE,1);%dilate
    imOut = imsubtract(imIn,openImage);%topHat
end

end
```

```
Error using topBottomHat (line 14)
Not enough input arguments.
```

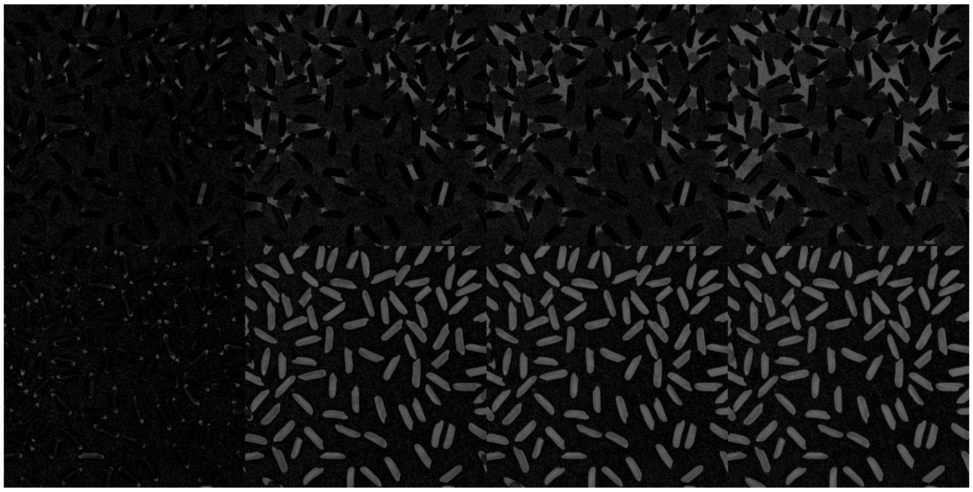
Published with MATLAB® R2014a

Problem 2 - continued

```
%Pratap Luitel
%ENGS 111
%HW 4


im1 = imread('rice.png');
[i2,m2] = imread('son1-1.gif');

%convert to type double
imIn1 = im2double(im1);
imIn2 = ind2gray(i2,m2);
imIn2 = im2double(imIn2);

imageSequence1 = zeros(size(imIn1,1),size(imIn1,2),1,8);
imageSequence2 = zeros(size(imIn2,1),size(imIn2,2),1,8);

for i = 1:4
    n = [7 11 13 15];

    startX = floor(n(i)/2);
    x = repmat(-startX:startX,[n(i) 1]);
    y = repmat([-startX:startX]',[1 n(i)]);
    r = sqrt(x.^2 + y.^2);

    SE = zeros(n(i),n(i));
    SE(r <= startX) = 1;

    %rice.png
    imageSequence1(:,:,1,i) = topBottomHat(imIn1,SE,0); %bottomHat
    imageSequence1(:,:,1,i+4) = topBottomHat(imIn1,SE,1); %topHat

    %son1-1.gif
    imageSequence2(:,:,1,i) = topBottomHat(imIn2,SE,0); %bottomHat
    imageSequence2(:,:,1,i+4) = topBottomHat(imIn2,SE,1); %topHat

end
figure(1)
montage(imageSequence1,'size',[2,4]);
figure(2)
montage(imageSequence2,'size',[2 4]);
```

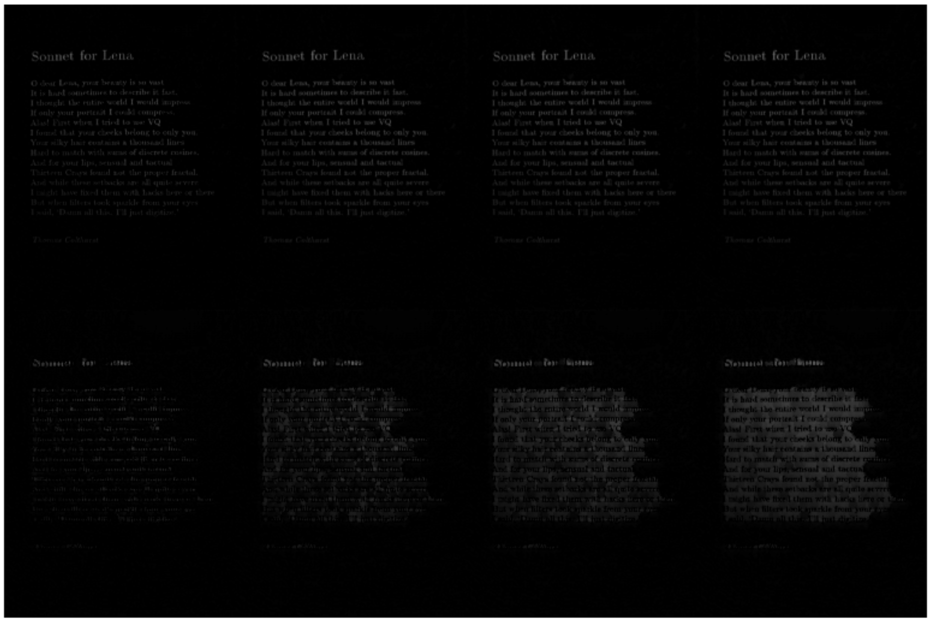
```
        Warning: Image is too big to fit on screen; displaying at 50%
```

Bottom Hat, size(SE) = 7 x 7 9 x 9 11 x 11 15 x 15



Top Hat, size(SE) = 7 x 7 9 x 9 11 x 11 15 x 15

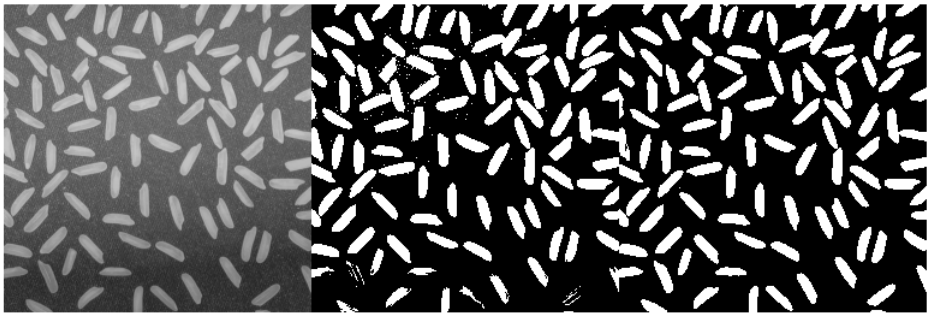
Bottom Hat, size(SE) = 7 x 7 9 x 9 11 x 11 15 x 15



Top Hat, size(SE) = 7 x 7 9 x 9 11 x 11 15 x 15

Published with MATLAB® R2014a

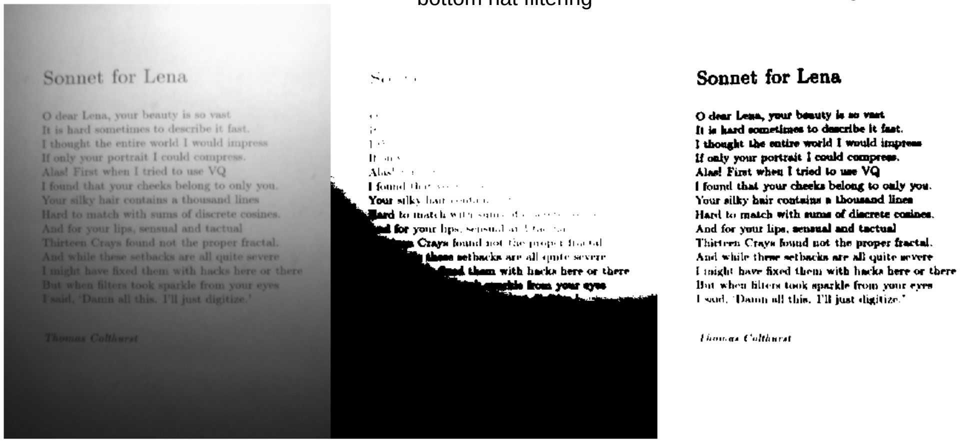
```
%Pratap Luitel
%ENGS 111
%HW 4
%Part 3


im1 = imread('rice.png');
[i2,m2] = imread('son1-1.gif');

%convert to type double
imIn1 = im2double(im1);
imIn2 = ind2gray(i2,m2);
imIn2 = im2double(imIn2);

imageSequence1 = zeros(size(imIn1,1),size(imIn1,2),1,3);
imageSequence2 = zeros(size(imIn2,1),size(imIn2,2),1,3);


n = [15];
%SE
startX = floor(n(1)/2);
x = repmat(-startX:startX,[n(1) 1]);
y = repmat([-startX:startX]',[1 n(1)]);
r = sqrt(x.^2 + y.^2);
SE = zeros(n(1),n(1));
SE(r <= startX) = 1;

%rice.png
imageSequence1(:,:,1,1) = imIn1;
%threshold without top hat filtering
imageSequence1(:,:,1,2) = im2bw(imIn1,0.5);
%threshold after top hat filtering
tempImIn1 = topBottomHat(imIn1,SE,1);
imageSequence1(:,:,1,3) = im2bw(tempImIn1,0.2);

%son1-1.gif
imageSequence2(:,:,1,1) = imIn2;
%threshold without bottom hat filtering
imageSequence2(:,:,1,2) = im2bw(imIn2,0.4);
%threshold after bottom hat filtering
tempImIn2 = topBottomHat(imIn2,SE,0);
imageSequence2(:,:,1,3) = ~im2bw(tempImIn2,0.06);


figure(1)
montage(imageSequence1,'size',[1,3]);
figure(2)
montage(imageSequence2,'size',[1 3]);
```



Original Image

Thresholding(>0.5) without TopHat filtering

Thresholding(>0.2) after topHat filtering



Original Image

Thresholding(> 0.4) without bottom hat filtering

Thresholding (< 0.6)after bottom hat filtering

Sonnet for Lena

O dear Lena, your beauty is so vast
It is hard sometimes to describe it fast.
I thought the entire world I would impress
If only your portrait I could compress.
Alas! First when I tried to use VQ
I found that your cheeks belong to only you.
Your silky hair contains a thousand lines
Hard to match with sums of discrete cosines.
And for your lips, sensual and tactual
Thirteen Crays found not the proper fractal.
And while these setbacks are all quite severe
I might have fixed them with hacks here or there
But when filters took sparkle from your eyes
I said, 'Damn all this. I'll just digitize.'

Thomas Colthurst

Published with MATLAB® R2014a