# Table of Contents

```
clear
close all

lpix=0.135;
lsize=round(5*(7/lpix)); % Average over 5 fiber-diameter length

FileName='Carbon_Hyfisyn_100';
O = imread([FileName,'.tif']);
O = uint8(O/256); % reducing memory use
%figure, imshow(O), title('Initial Image')
```

# Rotate image

```
angle=atand(200/27450)
O1 = imrotate(O,angle,'bilinear','crop');
%figure,imshow(O1),title('Rotated image')
```

```
angle =

    0.4174
```

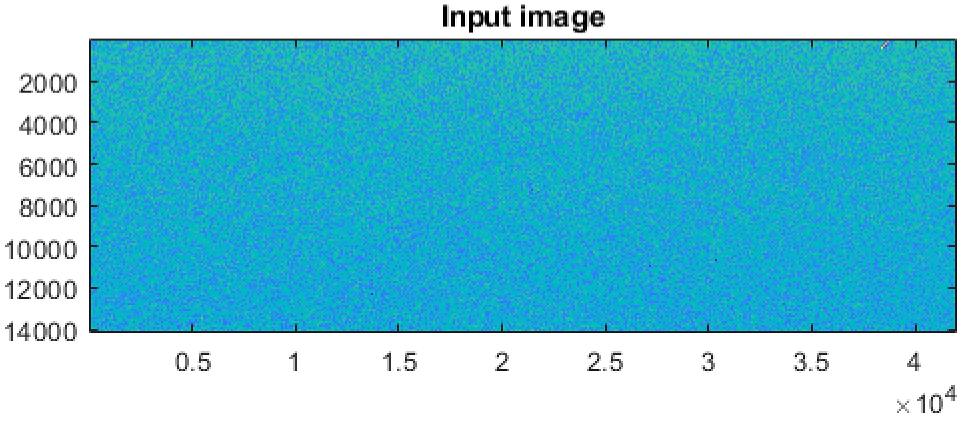
# Crop of rotate image for valid data

```
I1 = O1(153:14530,61:41980);
%figure,imshow(I1),title('Cropped rotated image')
```

# Crop of origional image for valid data

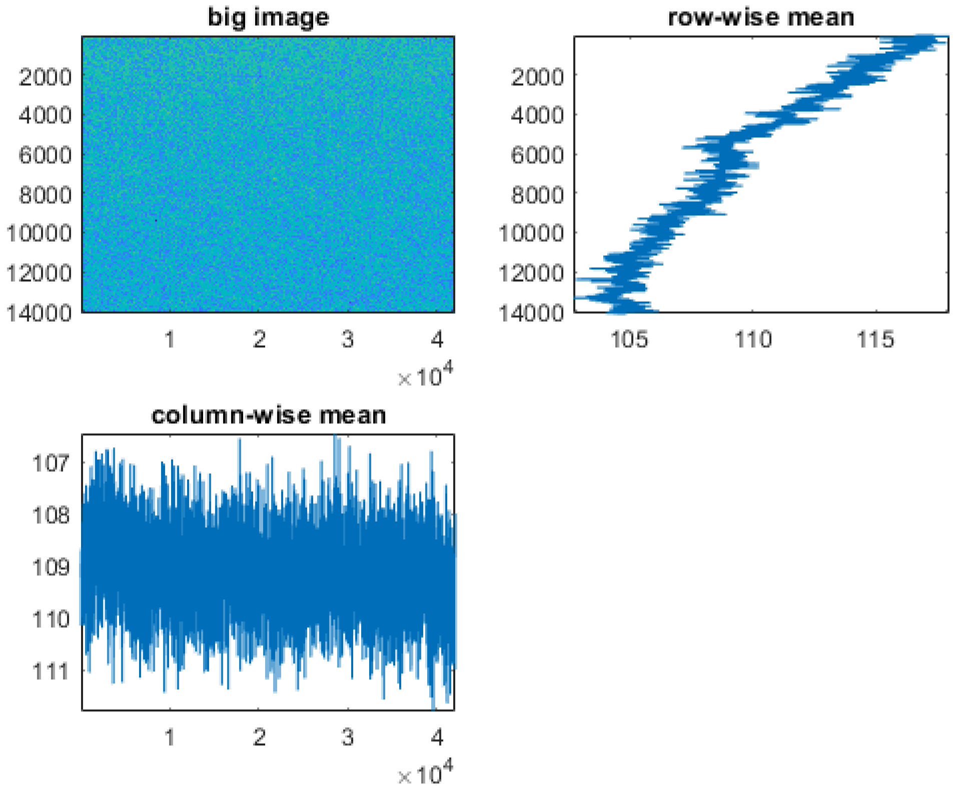
```
I2 = O(256:14380,:);
%figure,imshow(I2),title('Cropped origonal image')

I=I2;
```

```
figure, imagesc(I), axis image, title('Input image')
saveas(gcf,[FileName,'-AllI.png'])
saveas(gcf,[FileName,'-AllI'],'epsc')
savefig([FileName,'-AllI.fig'])
```



# investigatin bias in the image

```
figure,
subplot(2,2,1), imagesc(I), title('big image')
subplot(2,2,2), plot(mean(I,2),1:size(I,1)), axis ij tight,
 title('row-wise mean')
subplot(2,2,3), plot(1:size(I,2),mean(I,1)), axis ij tight,
 title('column-wise mean')
```



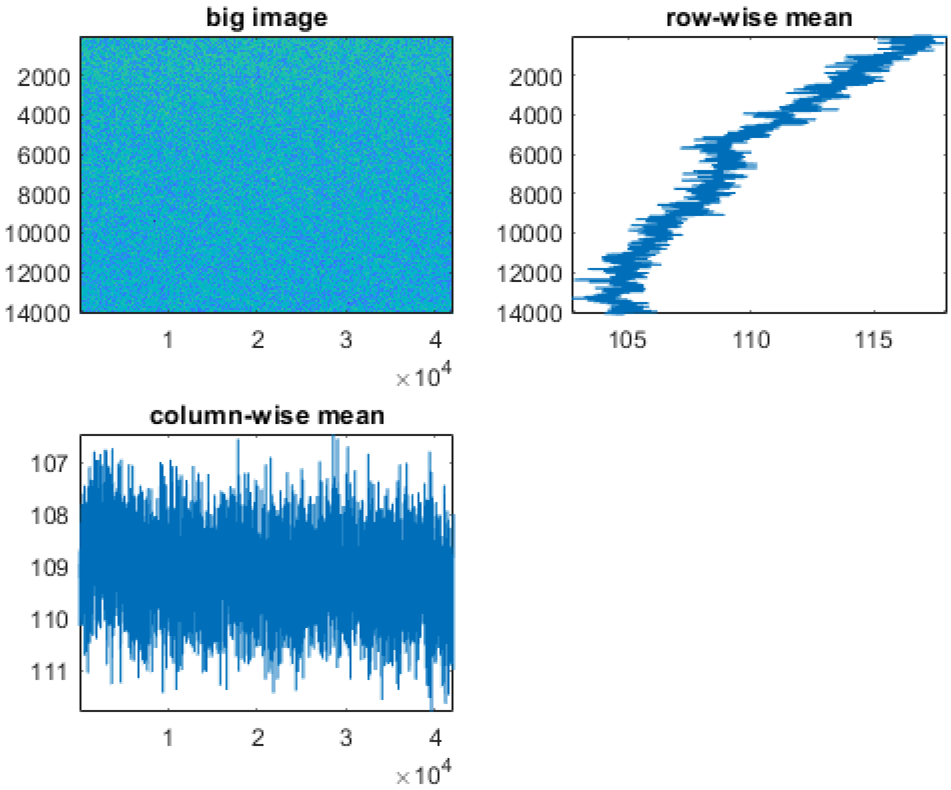
# cropping out

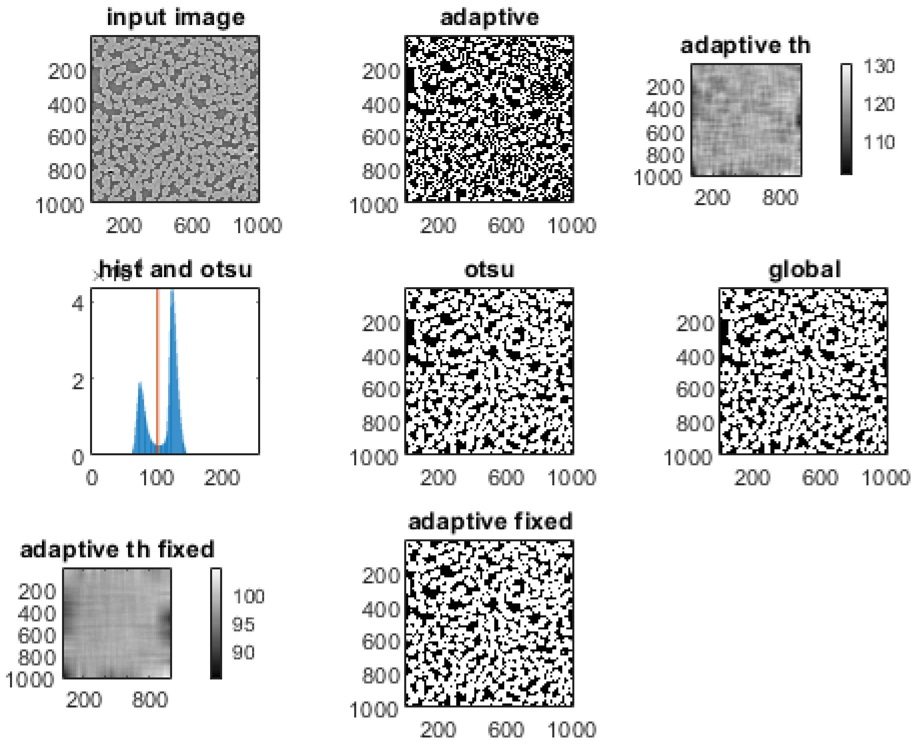
```
p = round(size(I)/2); % center of crop-out
r = 1000; % width of crop-out
C = I((1-r/2:r/2)+p(1),(1-r/2:r/2)+p(2)); % crop-out
```

# investigating different thresholding methods

```
h = hist(C(:),0:255);
th = otsuthresh(h)*255;

figure
subplot(3,3,1)
imagesc(C), axis image, colormap gray, title('input image')

subplot(3,3,2)
BW_adaptive = imbinarize(C, 'adaptive');
imagesc(BW_adaptive), axis image, title('adaptive')

T = adaptthresh(C);
subplot(3,3,3), imagesc(T*255), colorbar, axis image, title('adaptive th')

subplot(3,3,4)
bar(0:255,h), hold on, plot(th*[1,1],[0,max(h)]), axis tight square
```

```
title('hist and otsu')

subplot(3,3,5)
imagesc(C>th), axis image, title('otsu')

subplot(3,3,6)
BW_global = imbinarize(C, 'global');
imagesc(BW_global), axis image, title('global')

T = adaptthresh(C,0.7,'NeighborhoodSize',[301,301]);
subplot(3,3,7), imagesc(T*255), colorbar, axis image, title('adaptive
 th fixed')
subplot(3,3,8), imagesc(C>T*255), axis image, title('adaptive fixed')
```





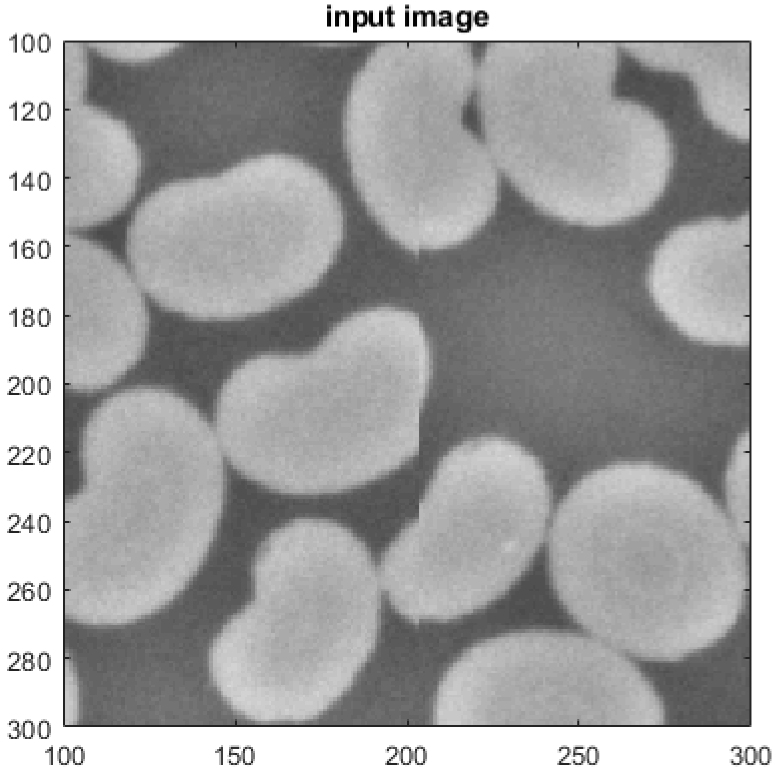
# Plot output

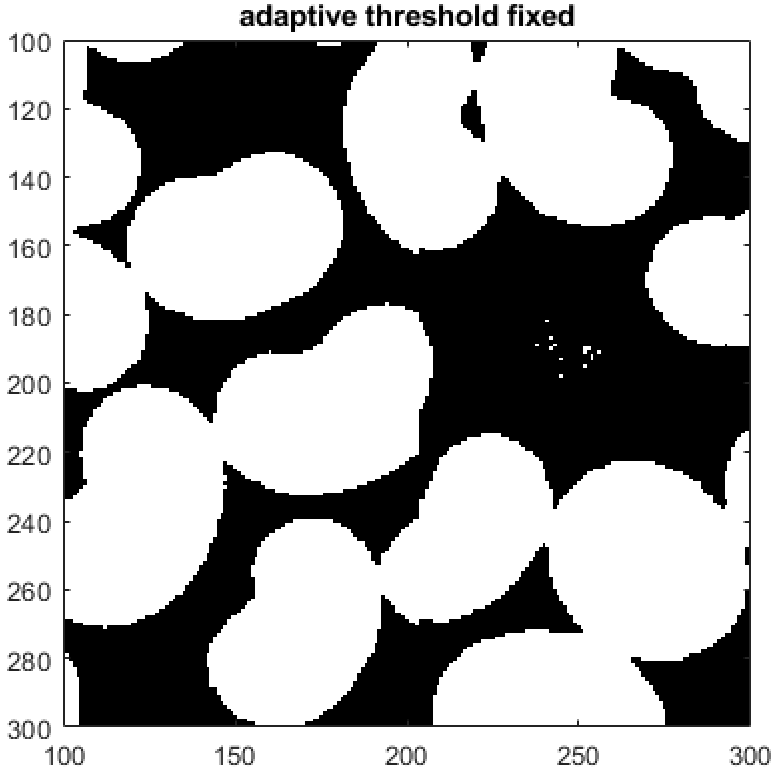
```
figure,
imagesc(C), axis image, ax=gca, ax.XLim = [100 300], ax.YLim = [100
 300], colormap gray, title('input image');
saveas(gcf,[FileName,'-zoomin.png'])
saveas(gcf,[FileName,'-zoomin'],'epsc')
savefig([FileName,'-zoomin.fig'])
figure,
imagesc(C>T*255), axis image, ax=gca, ax.XLim = [100 300], ax.YLim =
 [100 300], colormap gray, title('adaptive threshold fixed');
saveas(gcf,[FileName,'-zoomout.png'])
saveas(gcf,[FileName,'-zoomout'],'epsc')
savefig([FileName,'-zoomout.fig'])
figure,
bar(0:255,h), hold on, plot(th*[1,1],[0,max(h)]), axis tight square,
 title('Otsu threshold');
saveas(gcf,[FileName,'-OtsuThres.png'])
saveas(gcf,[FileName,'-OtsuThres'],'epsc')
savefig([FileName,'-OtsuThres.fig'])
```

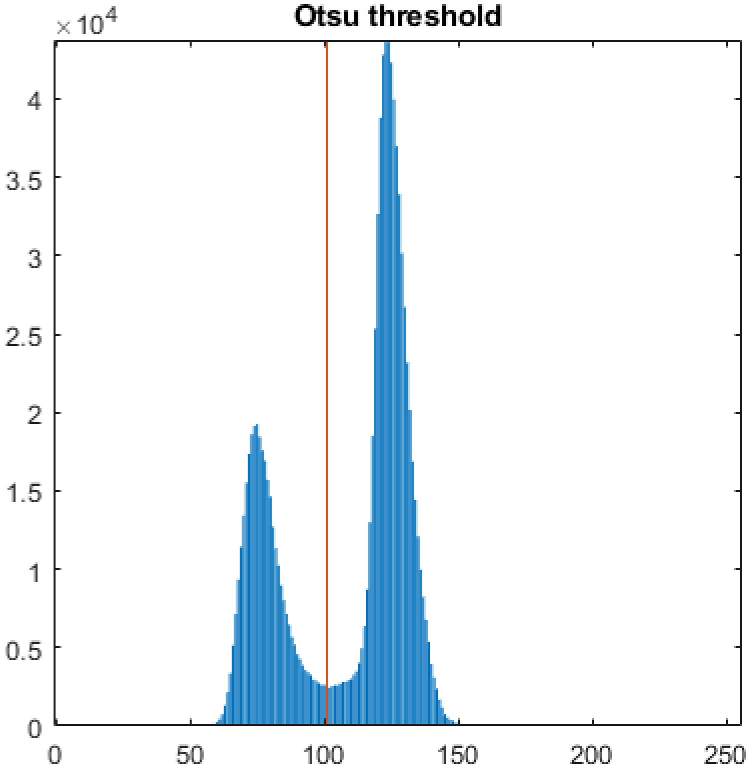
```
ax = 

  Axes with properties:
```

```
             XLim: [0.5000 1.0005e+03]
             YLim: [0.5000 1.0005e+03]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Use GET to show all properties


ax = 

  Axes with properties:

             XLim: [100 300]
             YLim: [0.5000 1.0005e+03]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Use GET to show all properties


ax = 

  Axes with properties:

             XLim: [100 300]
             YLim: [100 300]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Use GET to show all properties


ax = 

  Axes with properties:

             XLim: [0.5000 1.0005e+03]
             YLim: [0.5000 1.0005e+03]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Use GET to show all properties
```

```
ax = 

  Axes with properties:

             XLim: [100 300]
             YLim: [0.5000 1.0005e+03]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Use GET to show all properties


ax = 

  Axes with properties:

             XLim: [100 300]
             YLim: [100 300]
           XScale: 'linear'
           YScale: 'linear'
    GridLineStyle: '-'
         Position: [0.1300 0.1100 0.7750 0.8150]
            Units: 'normalized'

  Use GET to show all properties
```

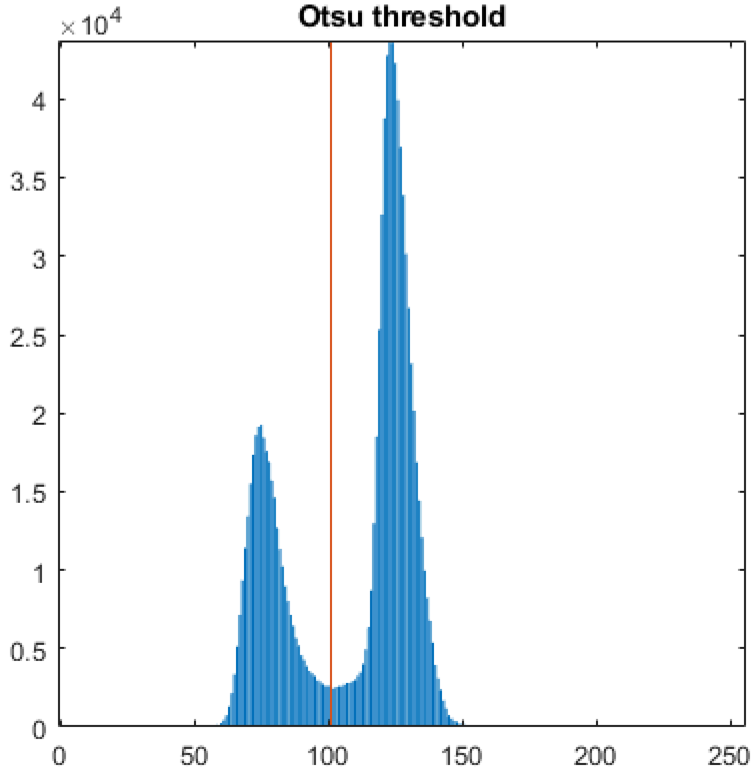

input image
100
120
140
160
180
200
220
240
260
280
300
100
150
200
250
300


adaptive threshold fixed
100
120
140
160
180
200
220
240
260
280
300
100
150
200
250
300



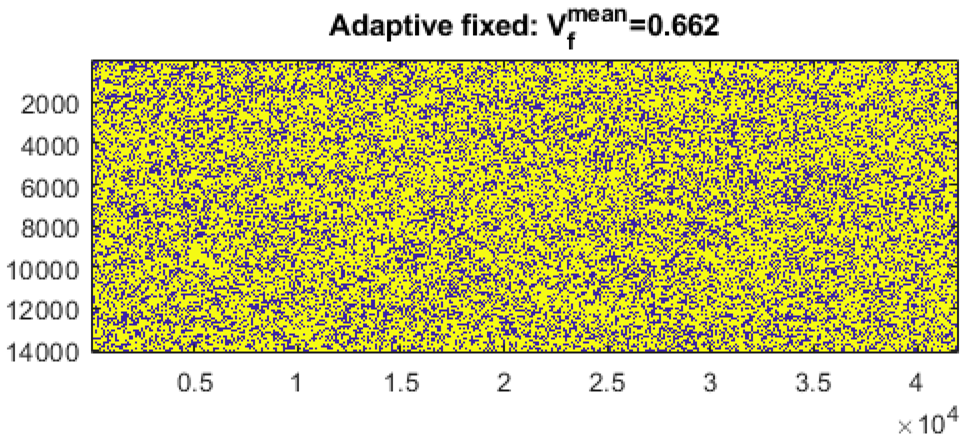
# whole image

```
T = adaptthresh(I,0.7,'NeighborhoodSize',[301,301]);
BW = I>T*255;
```

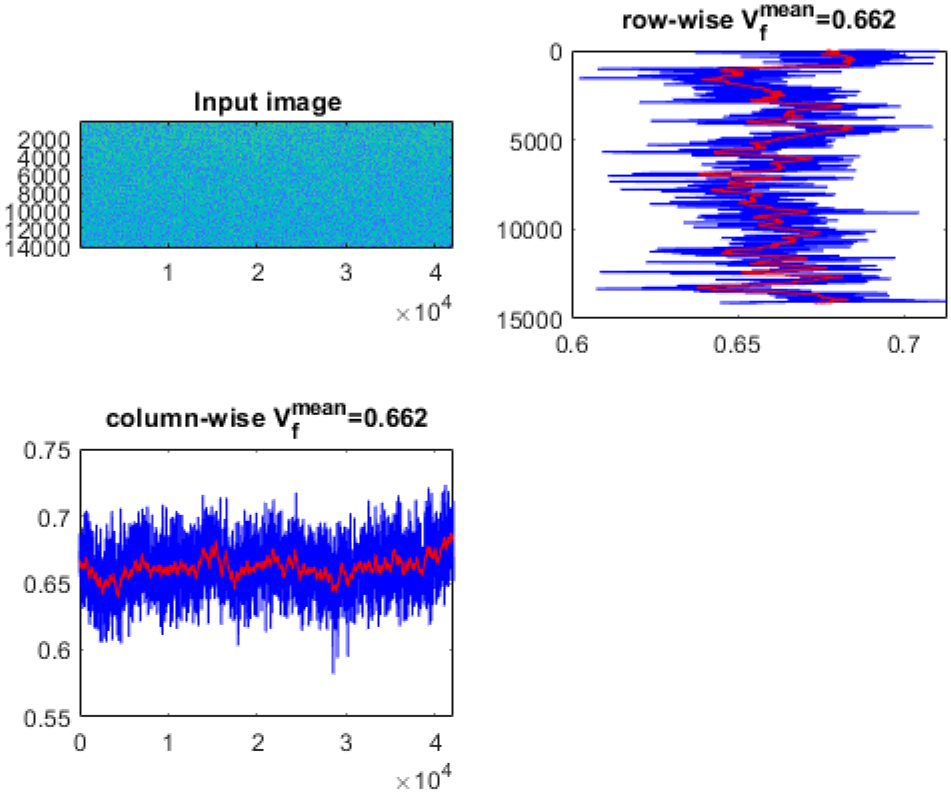


# Vf calculation

```
Vf = sum(BW(:))/numel(BW(:));
Vf2=sum(BW,2)/size(BW,2); % Variation
Vf1=sum(BW,1)/size(BW,1); % Variation
StrVf=['V_{f}^{mean}=',num2str(Vf,3)];

figure, imagesc(BW), axis image, title(['Adaptive fixed: ',StrVf])
saveas(gcf,[FileName,'-AllBW.png'])
saveas(gcf,[FileName,'-AllBW'],'epsc')
savefig([FileName,'-AllBW.fig'])
```



```
figure,
subplot(2,2,1), imagesc(I), axis image, title('Input image')
subplot(2,2,3), plot(1:size(BW,2),Vf1,'-b'), hold on,
 plot((1:size(BW,2)),movmean(Vf1,lsize),'-r'), title(['column-wise
 ',StrVf])
subplot(2,2,2), plot((Vf2),(1:size(BW,1)),'-b'), hold on,
 plot(movmean(Vf2,lsize),(1:size(BW,1)),'-r'),  title(['row-wise
 ',StrVf])
set(gca, 'YDir','reverse')
saveas(gcf,[FileName,'-LocalVf.png'])
saveas(gcf,[FileName,'-LocalVf'],'epsc')
savefig([FileName,'-LocalVf.fig'])
```



*Published with MATLAB® R2019b*