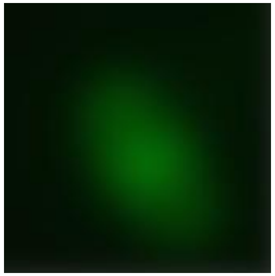
Week 10

Image Data Analysis(2)



Today
Open images with matlab
Reading image and video files
Thresholding
Convolution

Segmentation
1. Thresholding
2. Watershed

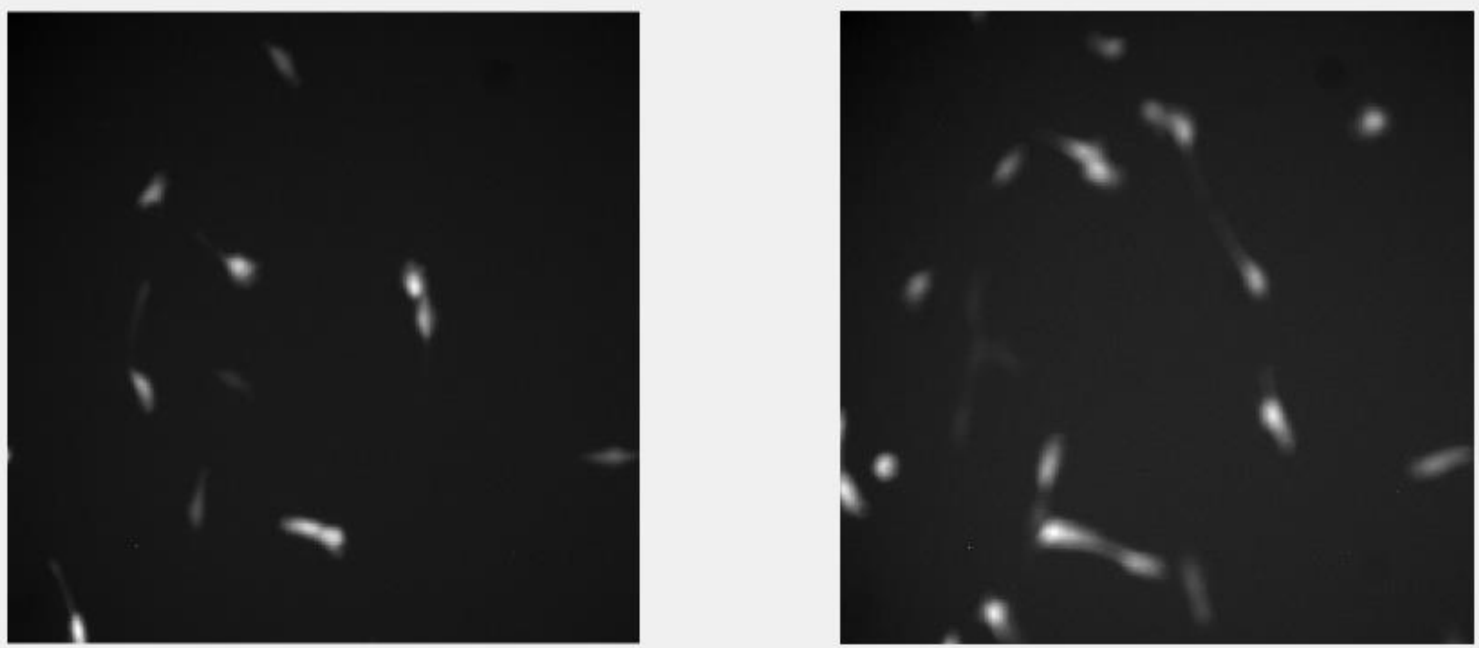
How to open image files?
Functions: dir, imfinfo

```
cd 'C:\Users\90539\Documents\Teaching\MBG5X\Exampledatasets_nurhan_KODMSON_new'

Spfile=dir('*.tif')
moivepro = imfinfo(Spfile(5).name);

xsize = moivepro(1).Width;
ysize = moivepro(1).Height;
imageData  = zeros(ysize, xsize, 1);
%%
```

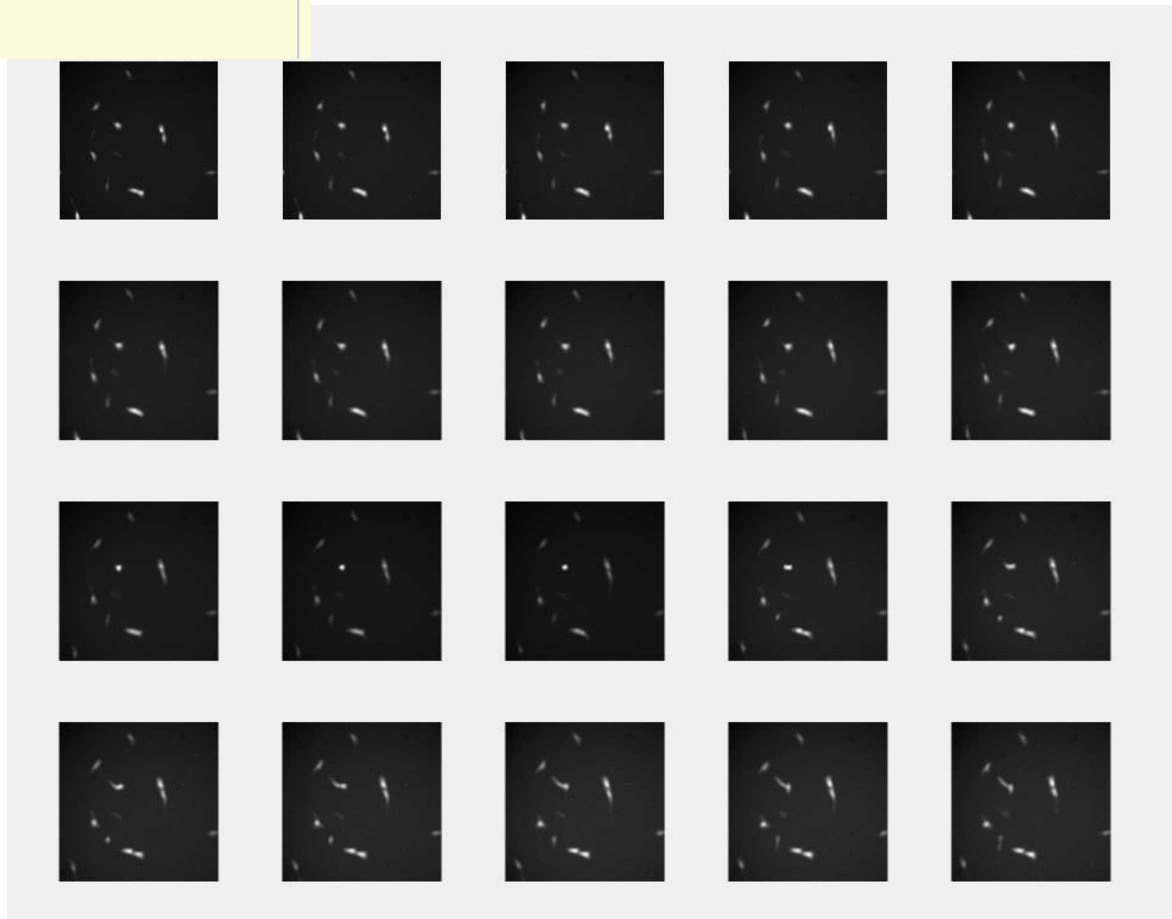
A single file have many image frames

```
%%
for i=1:100;
    fname= flistmovie{1};
    imageData(:,:,i) = imread(fname, 'tif', i);
end
%%
figure(1)
subplot(1,2,1)
imshow(imageData(:,:,1),[])
subplot(1,2,2)
imshow(imageData(:,:,80),[])
%%
```

Functions: imread



```
%%

figure(3)
for i=1:20;
    subplot(4,5,i)
imshow(imageData(:,:,i),[])
hold on
end
```

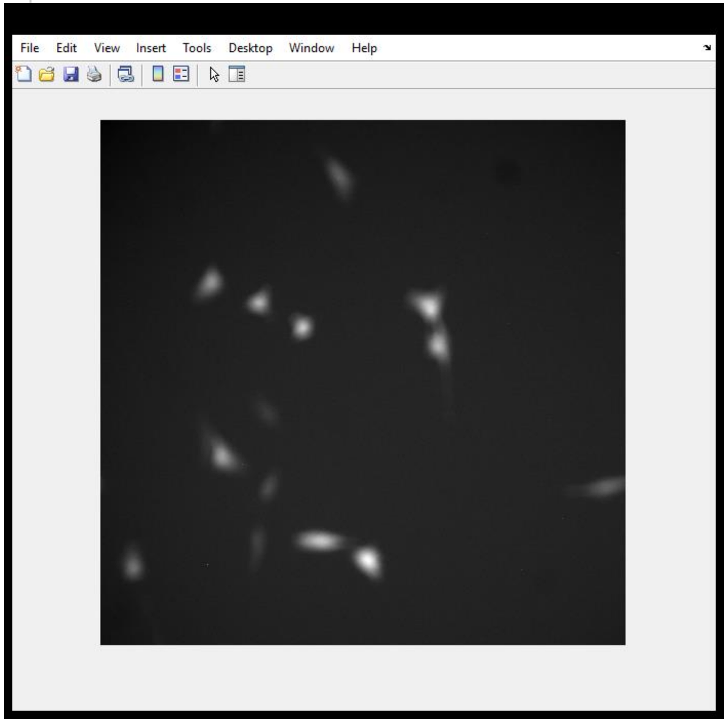


Insert the image file into cell array

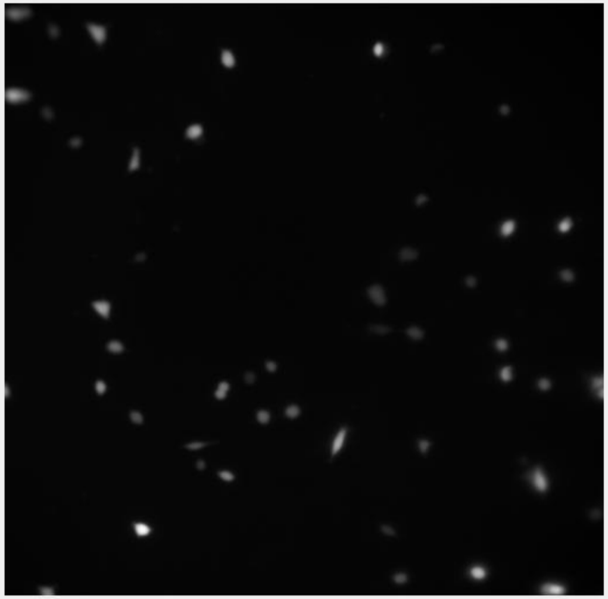
```
%%

imgLD{1}=double(imageData(:,:,1:80))
 smin=min(min(imgLD{1}(:,:,1)))
 smax=max(max(imgLD{1}(:,:,1)))
 sration=smax/smin
 [sr,sc,sl]=find(imgLD{1}(:,:,1)==smax)

 %%
```

Now run the movie file with the following code

```
 %%
  figure(1)
  for i=1:80

imshow(imgLD{1}(:,1:end,i),[])
hold on
pause(0.2)
 end
```



Open hela cell movies



Lecture_4_openimages_videos.m

```
% loadinf other movies
Spfile=dir('032 GFP.tif')
 flistmovie = {Spfile(1).name};

moivepro = imfinfo(flistmovie{1});
xsize = moivepro(1).Width;
ysize = moivepro(1).Height;
imageData  = zeros(ysize, xsize, 1);
imageData = imread( flistmovie{1}, 'tif', 1);
for i=1:40;

   fname= flistmovie{1};
   imageData(:,:,i) = imread(fname, 'tif', i);
end
%imgLD{1}=double(imageData)
imgLD{1}=double(imageData(:,:,1:40))
 smin=min(min(imgLD{1}(:,:,1)))
 smax=max(max(imgLD{1}(:,:,1)))
 sration=smax/smin
 [sr,sc,sl]=find(imgLD{1}(:,:,1)==smax)
 %

 for i=1:1
 figure(1)
imshow(imgLD{1}(:,1:end,1),[])
hold on
pause(0.5)
 end

%%
```

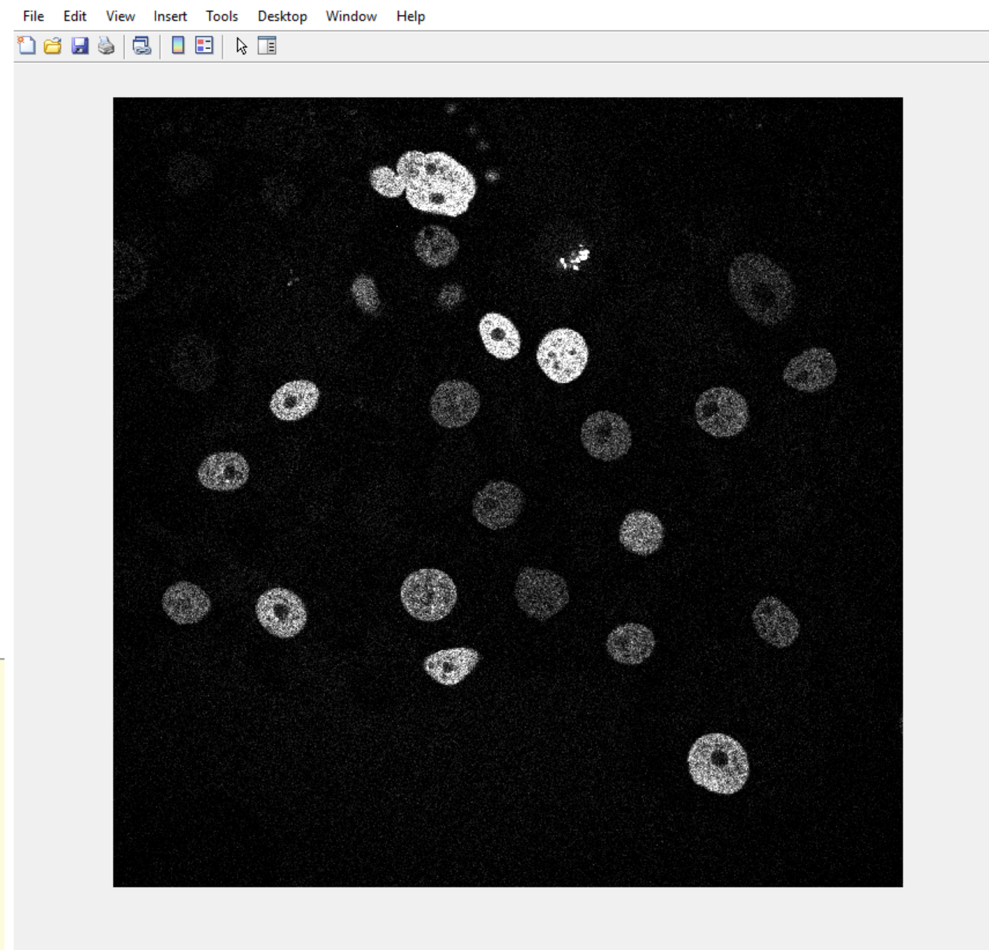
Sometimes movies files are saved as a individual image files

C: ▸ Users ▸ 90539 ▸ Documents ▸ Projects ▸ Software

Current Folder

Name	Size	Date Modified	Type
t000.tif	267 KB	11/23/2012 2:16 PM	TIF File
t002.tif	265 KB	11/23/2012 2:16 PM	TIF File
t003.tif	261 KB	11/23/2012 2:16 PM	TIF File
t004.tif	259 KB	11/23/2012 2:16 PM	TIF File
t005.tif	260 KB	11/23/2012 2:16 PM	TIF File
t006.tif	266 KB	11/23/2012 2:16 PM	TIF File
t007.tif	265 KB	11/23/2012 2:16 PM	TIF File
t008.tif	264 KB	11/23/2012 2:16 PM	TIF File
t009.tif	260 KB	11/23/2012 2:16 PM	TIF File
t010.tif	256 KB	11/23/2012 2:16 PM	TIF File
t011.tif	255 KB	11/23/2012 2:16 PM	TIF File
t012.tif	263 KB	11/23/2012 2:16 PM	TIF File
t013.tif	262 KB	11/23/2012 2:16 PM	TIF File
t014.tif	261 KB	11/23/2012 2:16 PM	TIF File
t015.tif	260 KB	11/23/2012 2:16 PM	TIF File
t016.tif	253 KB	11/23/2012 2:16 PM	TIF File
t017.tif	251 KB	11/23/2012 2:16 PM	TIF File
t018.tif	259 KB	11/23/2012 2:16 PM	TIF File

Details

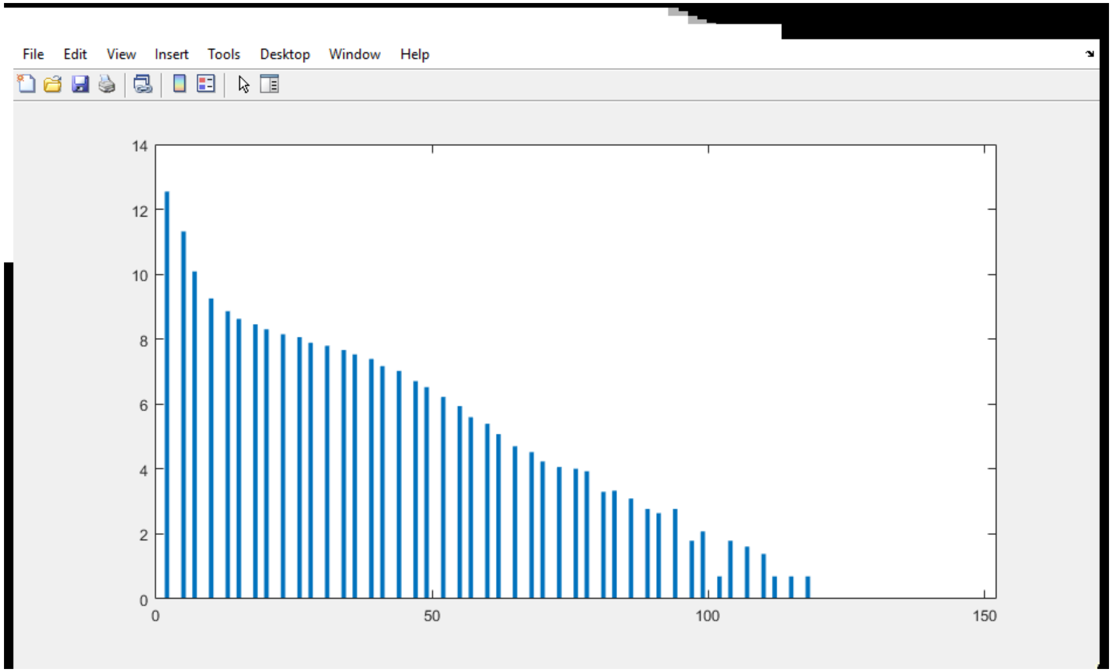



```
%%
for i=1:nframes;
    % read the name
    imageData(:,:,i) = imread(Spfile(i).name, 'tif', 1);
end

imgLD{1}=double(imageData);

 figure(1)
imshow(imgLD{1,1}(:,:,40),[min(min(imgLD{1,1}(:,:,1))) max(max(imgLD{1,1}(:,:,1)))/4.0])
max(max(imgLD{1,1}(:,:,1)))
min(min(imgLD{1,1}(:,:,1)))
saveas(gcf,'originalfiguresim_011.tif')
```

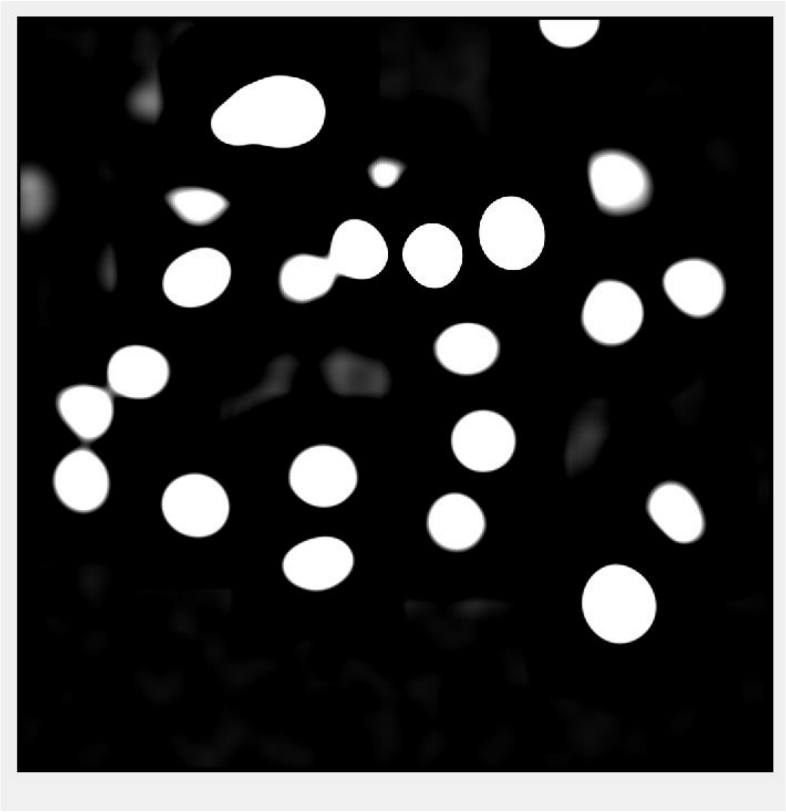
Image histograms

Functions: reshape, find, histcounts

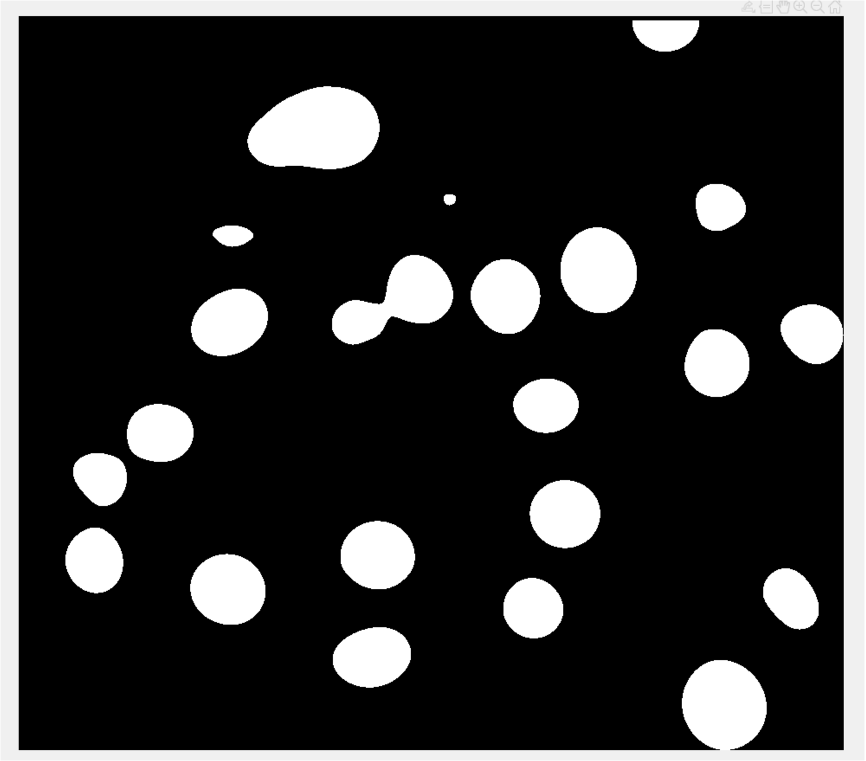
```
%%
x= reshape(imgLD{1}(:,:,1),1,[]);
y=sort(x,'ascend')

[r,c,l]=find(y>0)

edges=1:1:Lcutmax
n=histcounts(y(1,c(1,1):end),edges)

figure(2)
bar(edges(1:151), log(n))
```



Watershed based segmentation



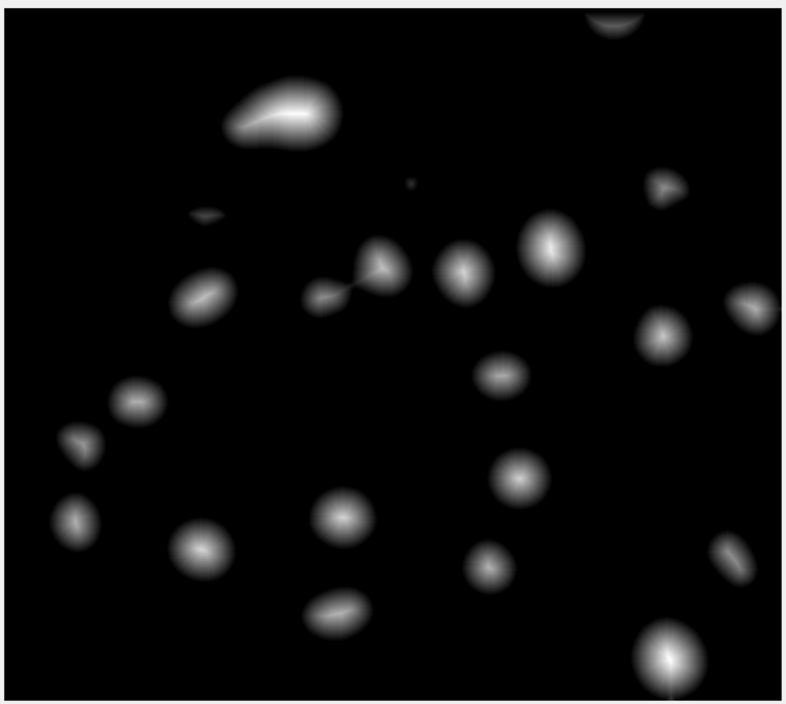
After convolution



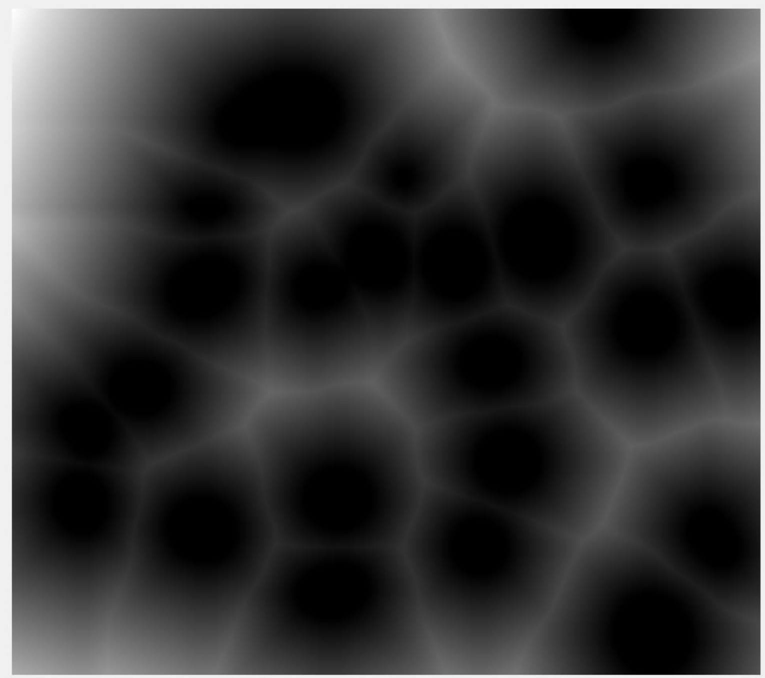
Binary image after thresholding

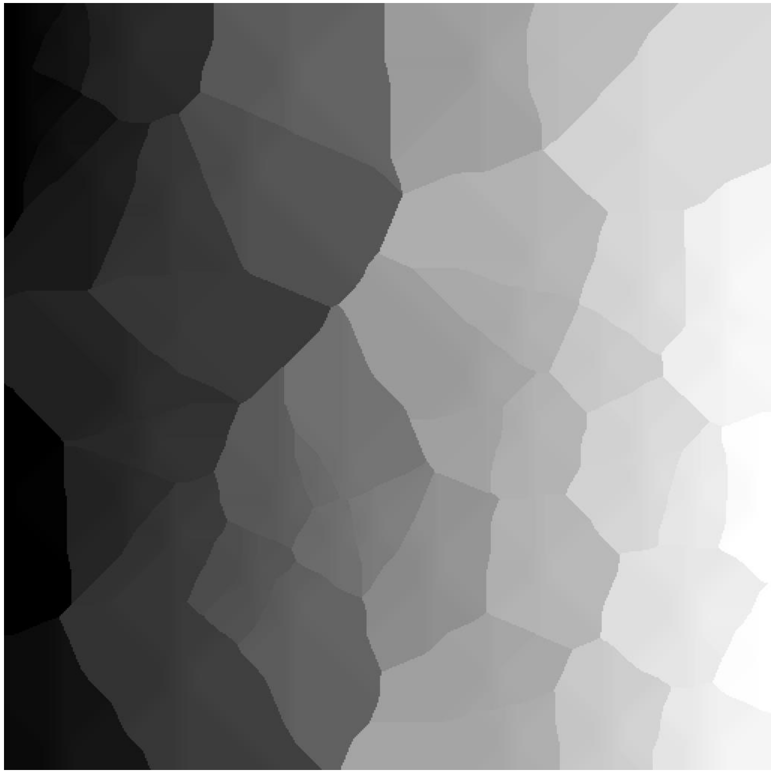
Bwdist, used for the distance transform of image, Distance between pixel and nearest nonzero pixel

```
%%
bw=imagetes;

D = bwdist(~imagetes,'quasi-euclidean');
%D = bwdist(~imagetes,'chessboard');
%figure(24)
%imshow(D,[])
%title('Distance Transform of Binary Image')
D = D;
figure(25)
imshow(D,[])
%%
```



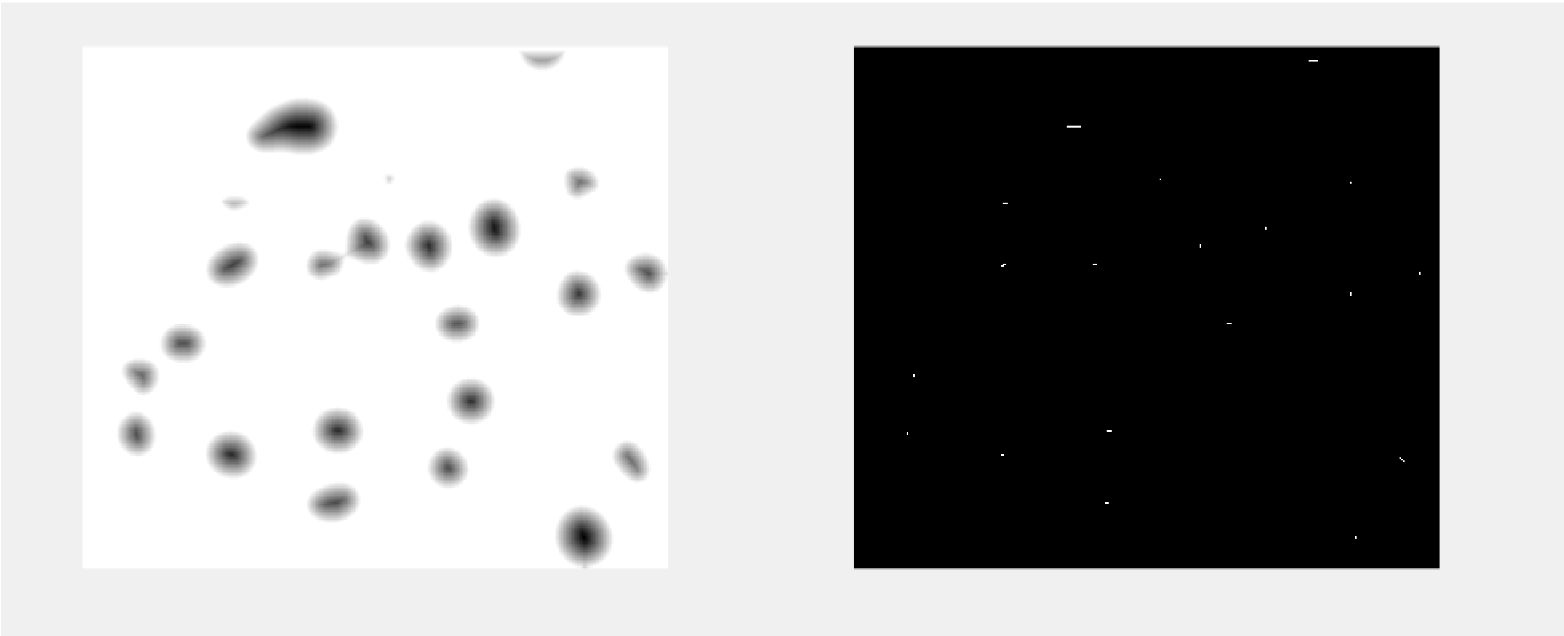
```
%%
bw=imagetes;

D = bwdist(imagetes,'quasi-euclidean');
%D = bwdist(~imagetes,'chessboard');
%figure(24)
%imshow(D,[])
%title('Distance Transform of Binary Image')
D = D;
figure(25)
imshow(D,[])
%%
```





```
%%
bw=imagetes;

[D,idx]  = bwdist(imagetes,'quasi-euclidean');
%D = bwdist(~imagetes,'chessboard');
%figure(24)
%imshow(D,[])
%title('Distance Transform of Binary Image')
Dx = -D;
figure(25)
subplot(1,2,1)
imshow(Dx,[])
subplot(1,2,2)
imshow(D,[])

figure(26)
imshow(idx,[])
```

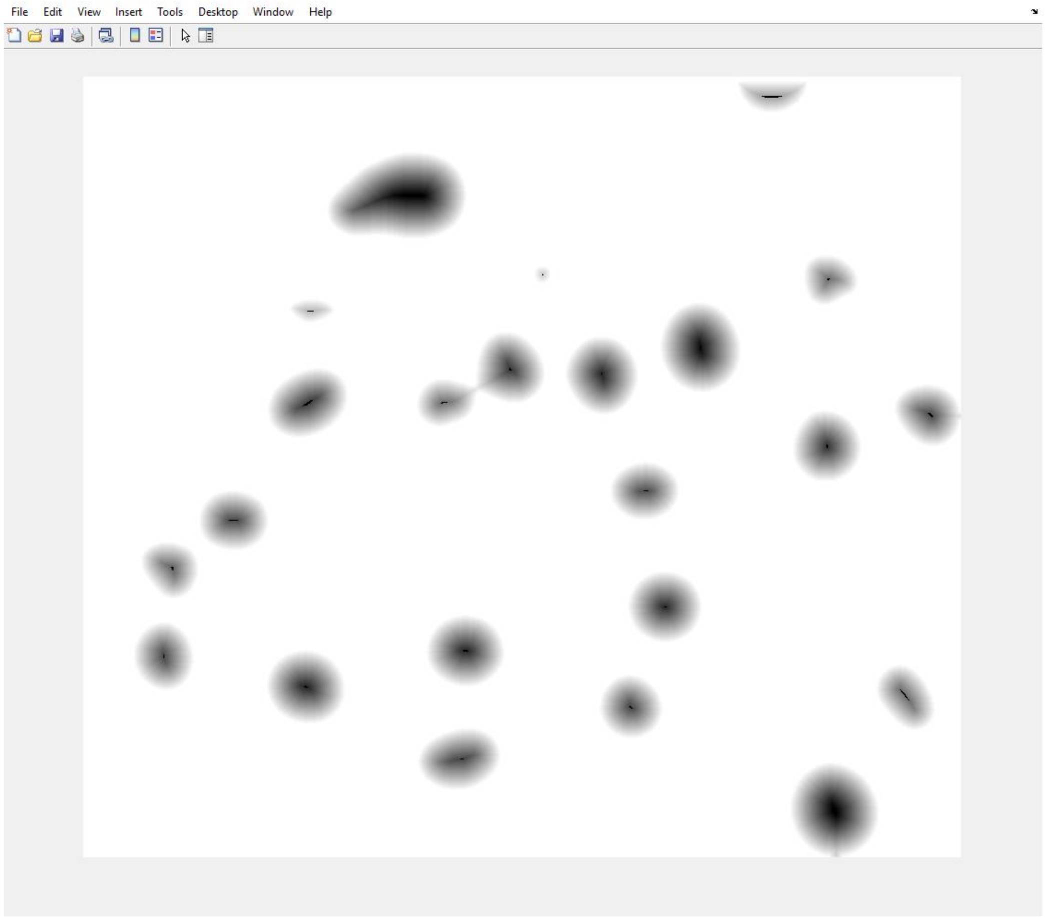
Here we compute the regional minima



```
%%
%title('Complement of Distance Transform')
%
%step 2
mask = imextendedmin( (-1*min(min(D)))*D,36);
% figure(34)
% imshowpair(bw,mask,'blend')
%
 figure(35)
 subplot(1,2,1)
 imshow(D,[])
 subplot(1,2,2)
 imshow(mask,[])
```

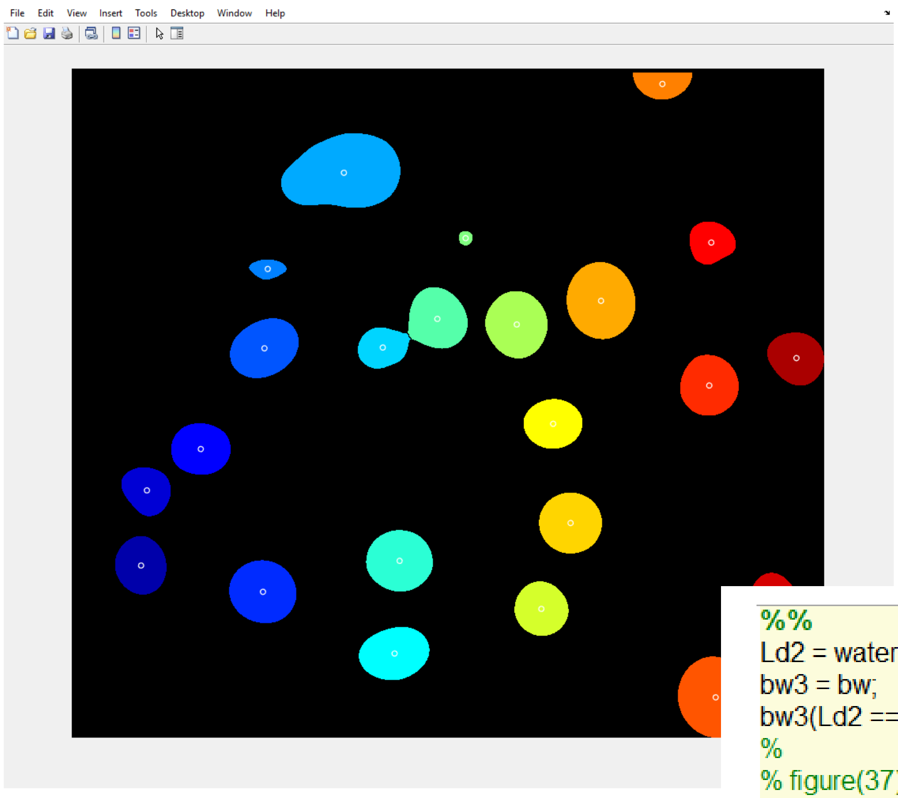
Here we find the global minimum for each region

Computation of marked image

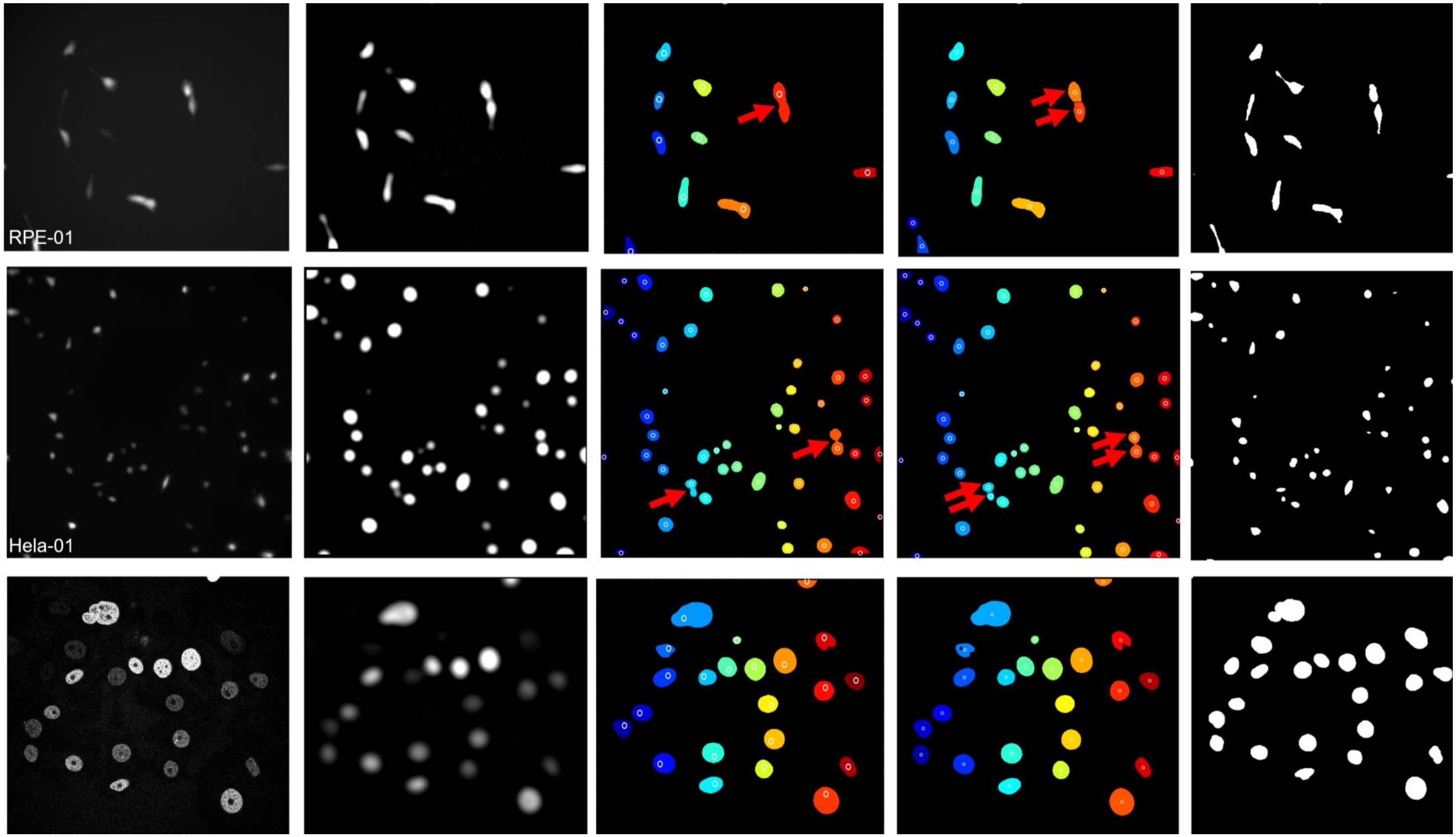



```
%%
D2 = imimposemin(D,mask);
figure(36)
 imshow(D2,[])
```

Here is the watershed image

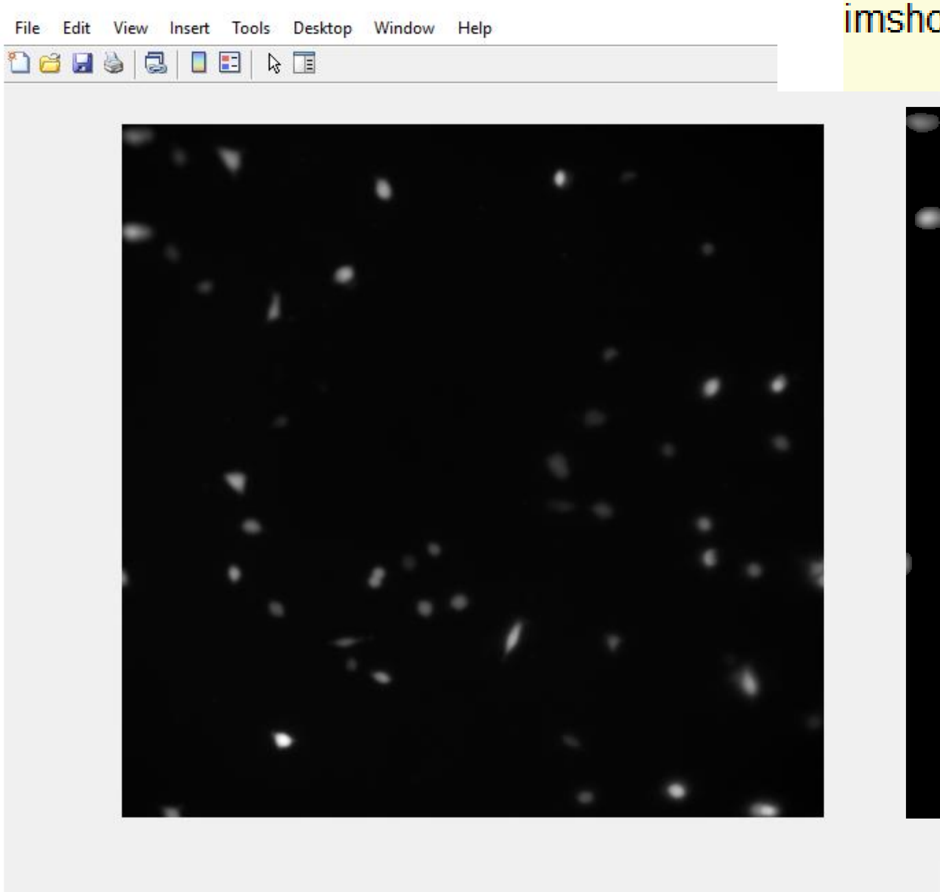


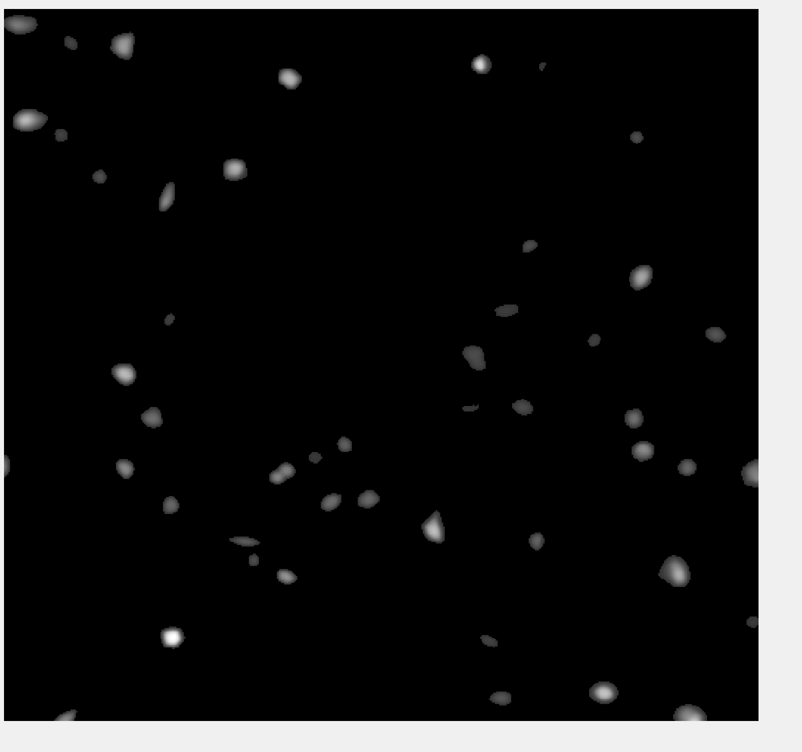

```
%%
Ld2 = watershed(D2);
bw3 = bw;
bw3(Ld2 == 0) = 0;
%
% figure(37)
% subplot(1,2,1)
% imshow(bw3,[])
% subplot(1,2,2)
% imshow(Ld2,[])

label1 = bwlabel(bw3);
rgb = label2rgb(label1,'jet',[0 0 0]);

stats = regionprops("table",label1,"Centroid",  "MajorAxisLength","MinorAxisLength")
figure(3)
imshow(rgb)
hold on
plot(stats.Centroid(:,1), stats.Centroid(:,2),'ow','linewidth',1,'Markersize',5)
```


RPE-01
Hela-01

Analysis of Hela Cells

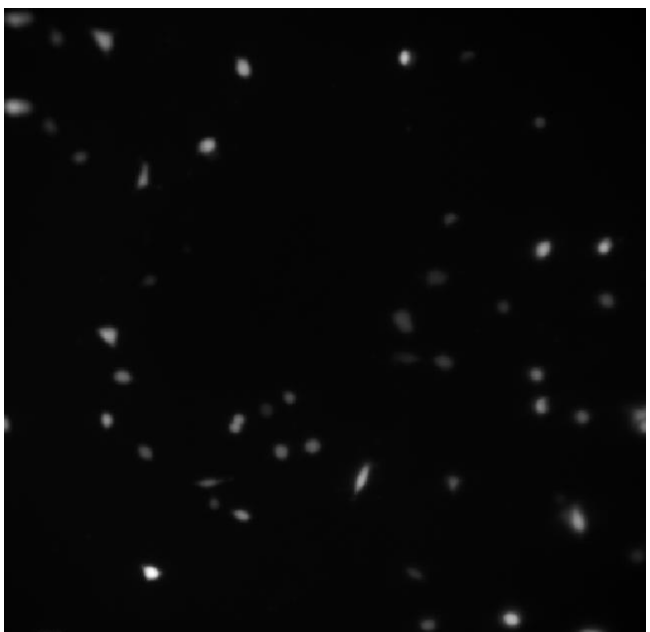
```
for i=1:nframes;
Lcut=min(min(imgLDout{i,1}));
background=Lcut*2
imgLDout{i,1}(imgLDout{i,1}<2000)=0;
end
figure(9)
imshow(imgLDout{3,1},[])
```





Convoluted image

```
%%
clear imageOut3
objectSize=50;
 noiseSize=3
for i=1:nframes;

   imageIn = imgLDout{i,1};
 [imageOut3{i,1}]=objFilter5son(imageIn,objectSize, noiseSize);
end

for i=1:1
figure(100)
subplot(1,2,1)
imshow(imgLD{1,1}(:,:,i),[])
subplot(1,2,2)
imshow(imageOut3{i,1}(:,:),[])

end
```



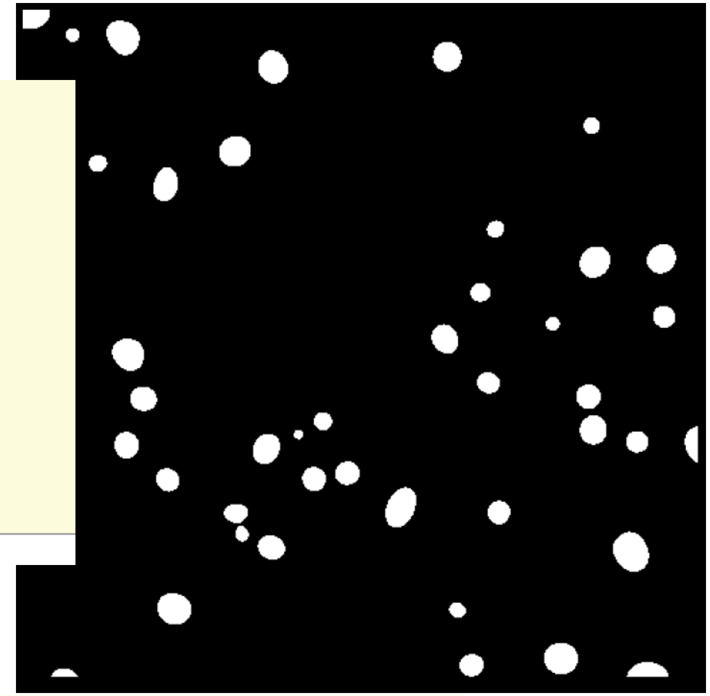
conv2

It is usd to remove pixel noise and create circular objects

```
function [imageOut]=objFilter(imageIn,objectSize,noiseSize)

% Software name: objFilter
%
% Software summary: To remove background pixel noise in time-lapse video frames
% using bandpass filter
%
% Brief description: This function can be used to apply bandpass filter for
% each image in video frames. The smoothed images are generated after remove pixel noise
% in images. To remove pixel noise, a gauss filter is accordingly computed
% by using noise size. Then it was applied in both dimensions.
%
% How to run code?
% To use this function, type the following line into Editor,
% Enter Spfile that includes the name of individual frames in a movie file,
% then frame number is neede and click Run.
%
% for i=1:nframes;
%
%   imageIn = imageLoad1{1}(:,:,i);
%  [imageOut{i,1}]=objFilter(imageIn,objectSize, noiseSize);
%
%  end
% Inputs:
%
% 1. imageIn: Background subtracted 2D image frames
% 2. objectSize: An average length of objects in frames. The default value is 50 pixel (10x obje
%    User has to open a single image and measure the length of a few cells in
%    pixel. Enter an average cell size.
% 3. noiseSize: Average length of noise. Set the pixel value as an integer number. The default
%    User has to check a few frames and determine the average diamter of
```

Now we can segments the image by finging the connected pixels

```
%%
% check image before separate pixels
katdegeri=1.10
rang=katdegeri*std(imageOut3{1,1}(:));
[rPixel,cPixel,lo]=find(imageOut3{1,1}(:,:) > rang);
imagetes=zeros(1024,1024);
imagetes=zeros(512,512);
for i=1:size(rPixel,1);
    imagetes(rPixel(i,1),cPixel(i,1))=1;
end
figure(13)
imshow(imagetes,[])
%%
```



```
%%
% here compute the cell regions
% segmentation step based on colvoluted image
tic
clear output1
for i=1:1;
    i
    [output1{ 1,1},output1{1,2}]=objExplorer2son(imageOut3{i,1},katdegeri,i);  %028

end
toc
```

```
toc
%%
l=size(imageOut3{1,1},1)
m=size(imageOut3{1,1},2)
imagetest=zeros(l,m);
cellidx=output1;
kl=1
for ts=1:size(cellidx{1,1},1);

    % here we fill the array with cell numbers from 1 to n
    % we use the coordinates data from
for i=1:size(cellidx{1,2},1);

    if cellidx{1,2}(i,3)==ts;
        imagetest(cellidx{1,2}(i,1),cellidx{1,2}(i,2))=kl;
    end
end
    kl=kl+1
end

objSgmnt{1,1}=imagetest;
colMatList=[0,0,0];
colMatList(2:257,1:3)=jet(256);

figure(2)
imagesc(imagetest)
colormap(colMatList)
colorbar
hold on
plot(cellidx{1,1}(1:end,1), cellidx{1,1}(1:end,2),'ow','linewidth',1,'Markersize',5)
saveas(gcf,'sim1segmentedimage2.tif')
save('segmenteddatasim2','output1')
```