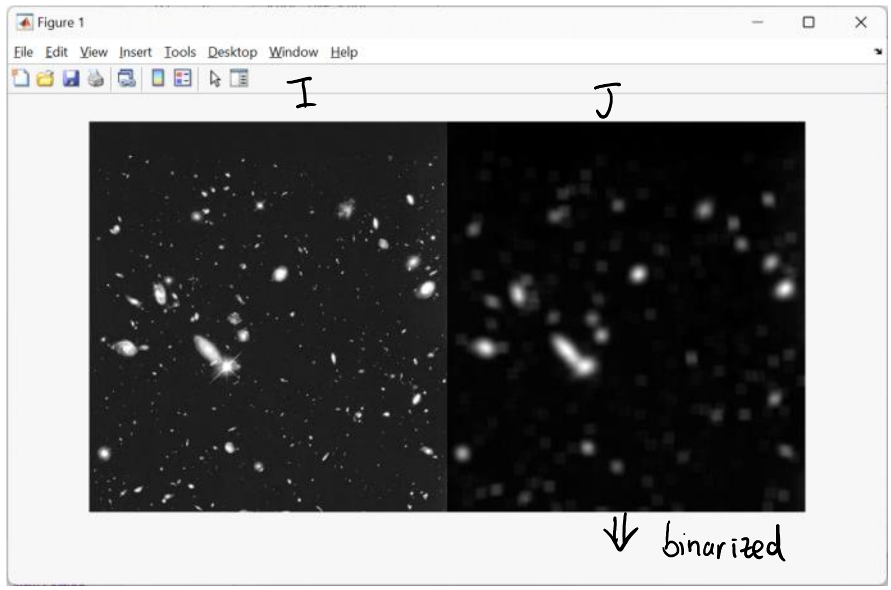
Lecture 15

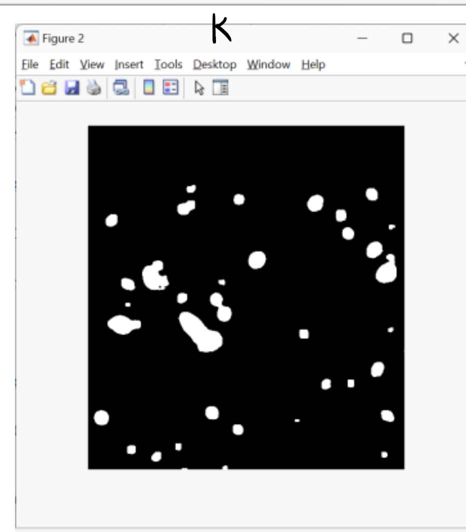
Spatial Filtering (cont'd)

Smoothing filter followed by thresholding to extract ROI's

```
>> I = imread('Fig0334(a)(hubble-original).tif');
>> imshow(I)
>> h = ones(15, 15)/(15^2);
>> J = imfilter(I, h, 'symmetric');
>> imshowpair(I, J, 'montage')
```

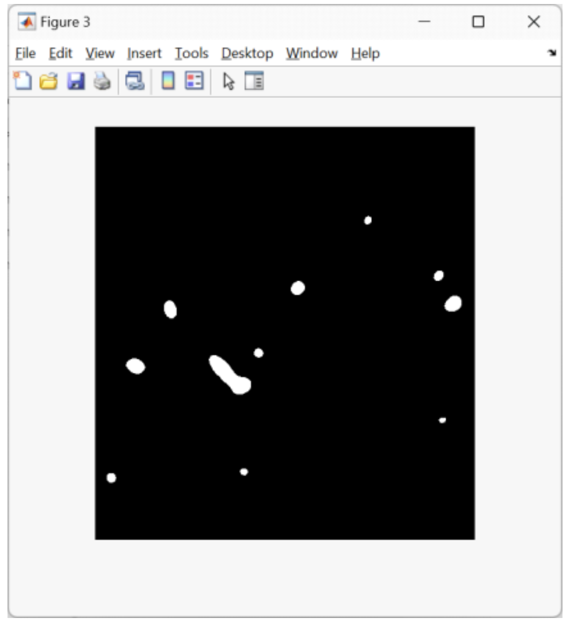



```
>> max(J(:))
ans =
  uint8
   219

>> Th = uint8(ratio*(double(max(J(:)))))
Th =
  uint8
   55

>> K = (J > Th);
>> figure; imshow(K, []);
```




```
>> max(K(:))
ans =
  logical
   1

>> min(K(:))
ans =
  logical
   0
```

```
>> ratio = 0.5;
>> Th = uint8(ratio*(double(max(J(:)))))

Th =

  uint8

   110

>> K = (J > Th);
figure; imshow(K, []);
```

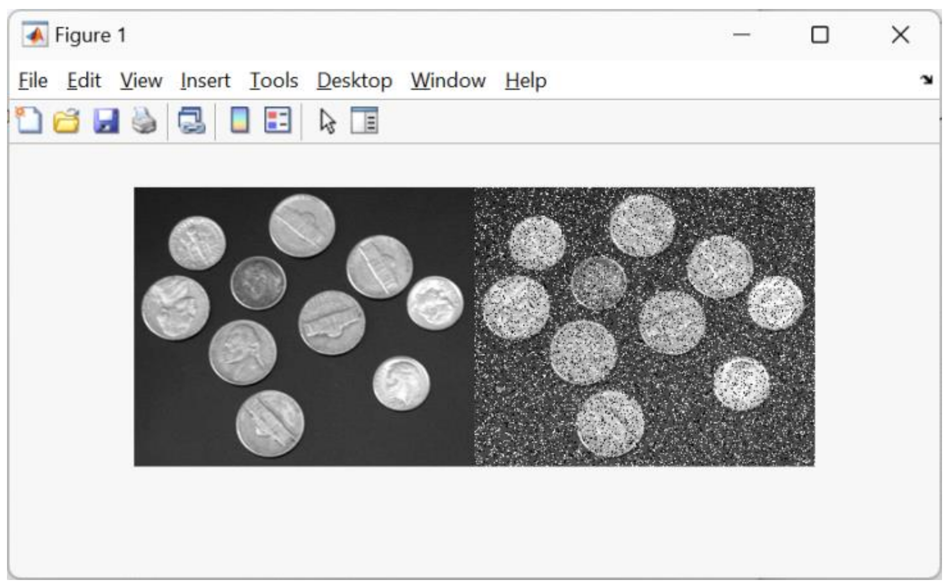


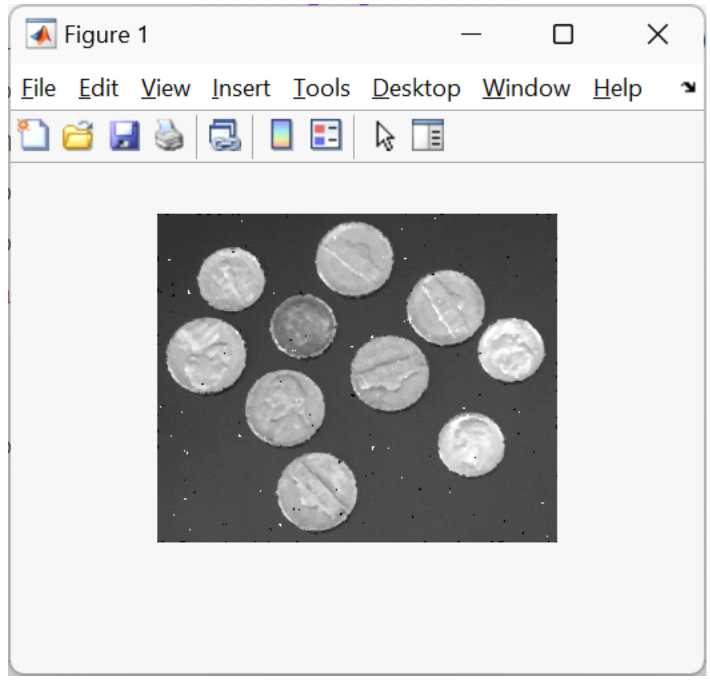
- Impulse Noise (Salt-and-Pepper Noise)

J = imnoise(I,'salt & pepper',d) adds salt and pepper noise, where d is the noise density. This affects approximately d*numel(I) pixels.

density

```
>> I = imread('coins.png');
>> J = imnoise(I, 'salt & pepper', 0.2);
>> imshowpair(I, J, 'montage')
```



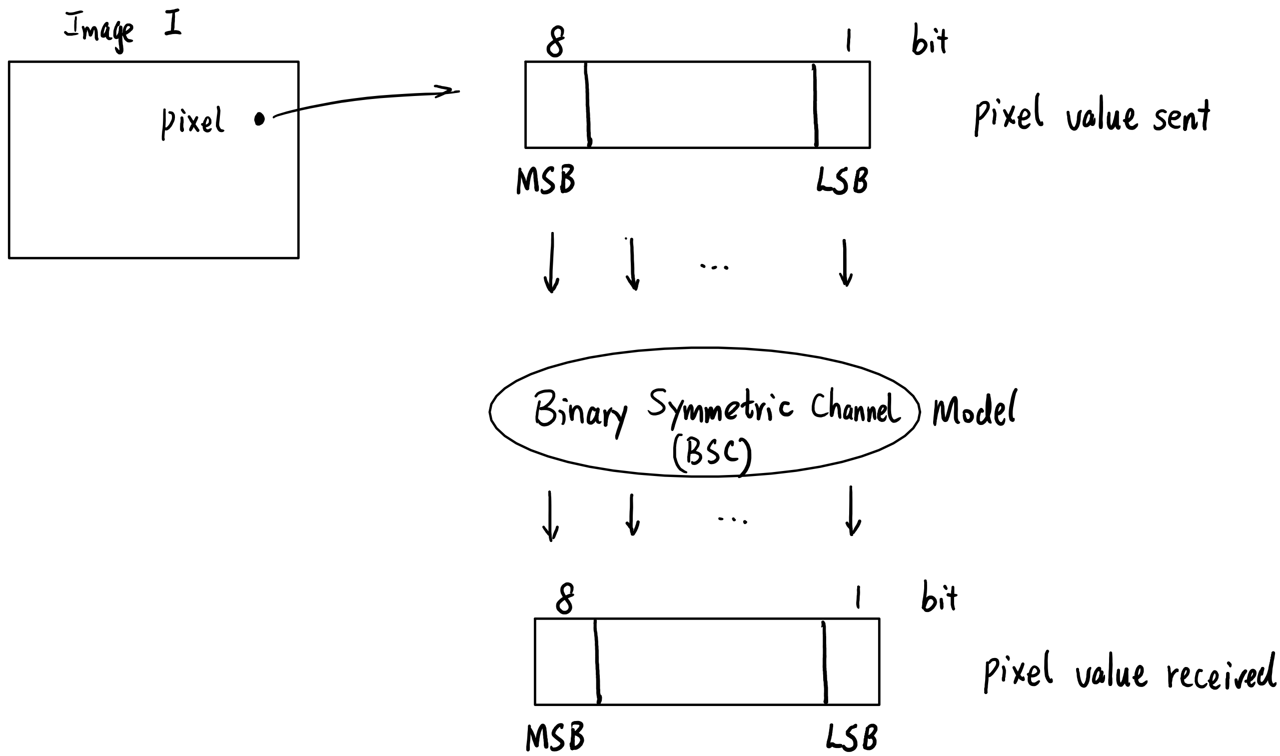


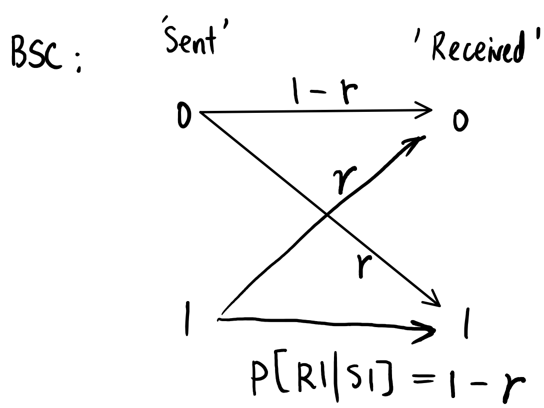
>> K = medfilt2(J);

>> edit imnoise

```
case 'salt & pepper'
  p3 = 0.05;   % default density

b = images.internal.algimnoise(a, code, classIn, classChanged, p3, p4);

function b = algimnoise(a, code, classIn, classChanged, p3, p4)
% Main algorithm used by imnoise function. See imnoise for more
% details

case 'salt & pepper' % Salt & pepper noise
    b = a;
    x = rand(sizeA);   % x is an random image with pixel value (0, 1)
    b(x < p3/2) = 0; % Minimum value    % 'pepper'
    b(x >= p3/2 & x < p3) = 1; % Maximum (saturated) value    % 'salt'
```

Impulse Noise Analysis (due to noisy communication links, or due to noisy sensors ….)





Cross-Over Probability (conditional probabilities)

$r = p[R1 | S0]$ ← condition

$= p[R0 | S1]$

Difference between a pixel value before and after its bits go through the channel (BSC)
First, look at the case where only 1 bit was flipped:

MSB: 0 -> 1, difference = (1 - 0) x 128 = 128 $= 2^7$
1 -> 0 , difference = (0 - 1) x 128 = -128
Squared Error (SE) = $(2^7)^2 = 2^{14}$

LSB: 0 ->1, difference = (1 - 0) = 1
1 -> 0 , difference = (0 - 1) = -1
Squared Error (SE) = 2^0

In general, for the input (original) image I, with pixel value

$$X = \sum_{i=0}^{B-1} b_i 2^i, \quad \text{where } B = 8$$

bit value $b_i = \begin{cases} 0 \\ 1 \end{cases}$, where $i = 0, 1, \cdots, B-1$

J : Received image, with pixel value Y

$$\text{Prob}\left[\; |X - Y| = 2^i = ? \; \right]$$

Assume that only the ith-bit was flipped.