

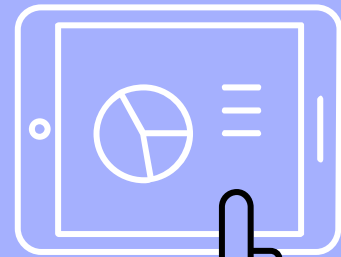


# MATLAB Matrix Operation

Local Image Process and Resolution adjustment



盧家鋒 Chia-Feng Lu, Ph.D.  
Department of Biomedical Imaging  
and Radiological Sciences, NYCU  
[alvin4016@nycu.edu.tw](mailto:alvin4016@nycu.edu.tw)



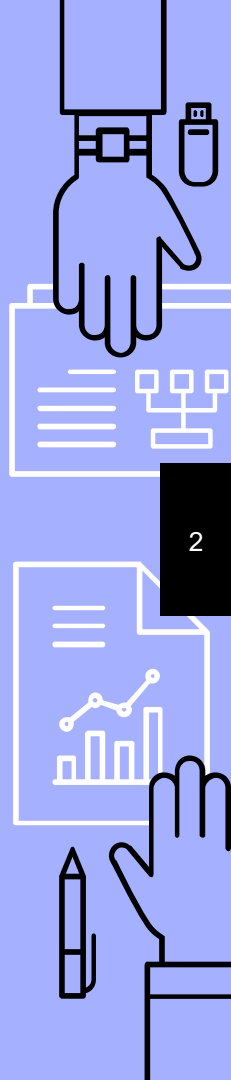


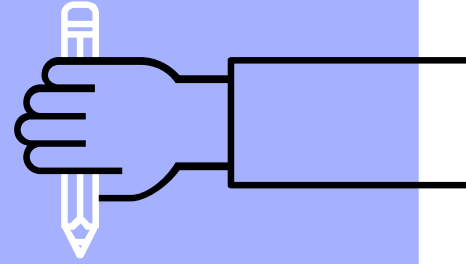
# Contents

- ▶ Matrix index and operation
- ▶ Local Image Process and Resolution adjustment

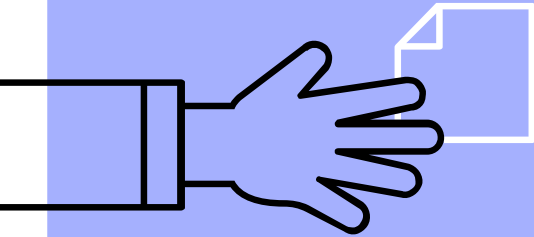
Please download the handout and materials from

[http://cflu.lab.nycu.edu.tw/CFLu\\_course\\_matlabimage.html](http://cflu.lab.nycu.edu.tw/CFLu_course_matlabimage.html)





# Matrix index and operation



# Format of Index

Create matrix A.

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

Column-wise index

Row-column index

```
Command Window
>> A(18)

ans =

6
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(3,4)

ans =

6
```

# Get Value

A specific range of index

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

Column-wise index

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

Row-column index

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
Command Window
>> A([12:14 17:19])

ans =

    4    64    6    6    6    7
```

```
Command Window
Command Window
>> A(2:4,3:4)

ans =

    4    6
   64    6
    6    7
.
```

# Get Value: Try it

A specific range of index

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

Column-wise index

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

Row-column index

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

# Delete Rows or Columns

**A**

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

```
Command Window
>> A(1:3,3:4)=[]
Subscripted assignment dimension mismatch.
```

Column-wise index

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

Row-column index

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(4,:)=[]

A =

    12     2    33     4    46
     3     7     4     6     5
     4     6    64     6    84
     7     4    45    23    78
```

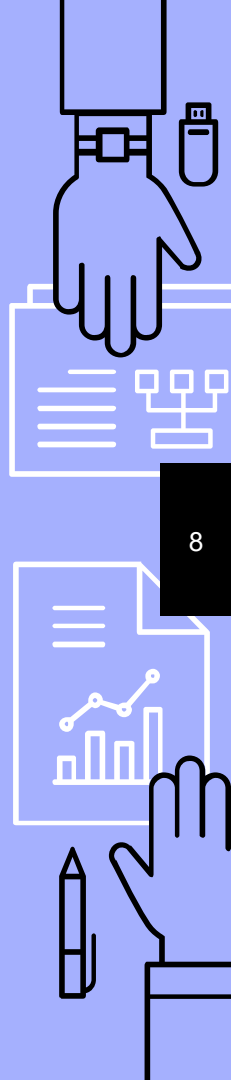
# Useful Function **help** it!

- ▶ **find**
  - Find indices of matched elements.
- ▶ `ind = find(A==6)`
  - **% get column-wise index**
- ▶ `[row, col] = find(A==6)`
  - **% get row-column index**

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

**A**

Try to find the indices with value larger than 10.





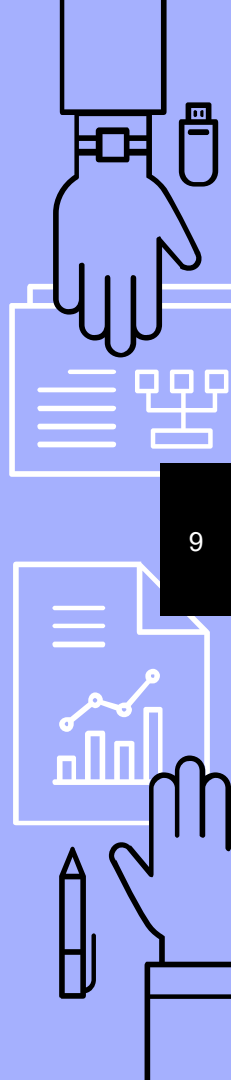
# Relation and Logical Operators

## % Relation operators

Equal	==
Not equal	~=
Less than	<
Greater than	>
Less than or equal	<=
Greater than or equal	>=

## % logical operators

And	&
Or	
Not	~



# Set Value

Rewrite values at specific locations.

A

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

Column-wise index

Row-column index

```
Command Window
>> A(7)=100

A =

    12     2    33     4    46
     3   100     4     6     5
     4     6    64     6    84
     6     5     6     7    37
     7     4    45    23    78
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(2,2)=100

A =

    12     2    33     4    46
     3   100     4     6     5
     4     6    64     6    84
     6     5     6     7    37
     7     4    45    23    78
```

# Set Value

- ▶ Replace value of 6 by 100
  - `ind = find(A==6);`
  - `A(ind) = 100;`

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

# Set Value

Rewrite values at specific locations.

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

Column-wise index

Row-column index

```
Command Window
>> A([12:14 17:19])=100

A =

12  2  33  4  46
 3  7 100 100  5
 4  6 100 100 84
 6  5 100 100 37
 7  4  45 23  78
```

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

```
Command Window
>> A(2:4,3:4)=100

A =

12  2  33  4  46
 3  7 100 100  5
 4  6 100 100 84
 6  5 100 100 37
 7  4  45 23  78
```

# Set Value

```
Command Window
>> A([12:14 17:19])=[100 102 104 101 103 105]

A =

12  2  33  4  46
 3  7 100 101  5
 4  6 102 103 84
 6  5 104 105 37
 7  4  45 23  78
```

Column-wise index

(1)	(6)	(11)	(16)	(21)
(2)	(7)	(12)	(17)	(22)
(3)	(8)	(13)	(18)	(23)
(4)	(9)	(14)	(19)	(24)
(5)	(10)	(15)	(20)	(25)

12	2	33	4	46
3	7	4	6	5
4	6	64	6	84
6	5	6	7	37
7	4	45	23	78

A

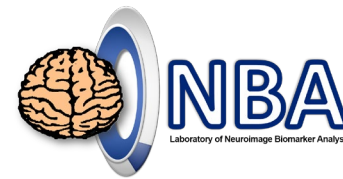
```
Command Window
>> A(2:4,3:4)=[100 101;102 103;104 105]

A =

12  2  33  4  46
 3  7 100 101  5
 4  6 102 103 84
 6  5 104 105 37
 7  4  45 23  78
```

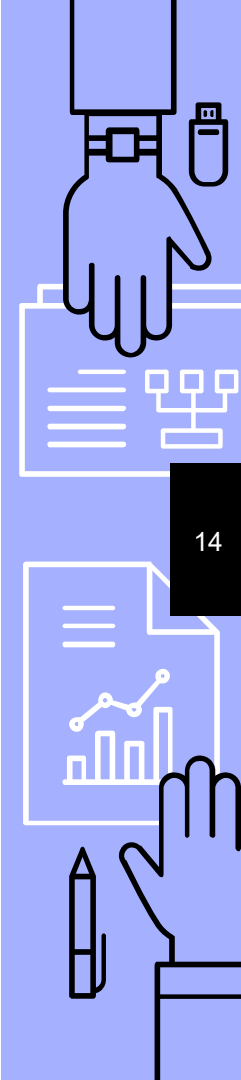
Row-column index

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)



# Useful Functions **help** it!

- ▶ **size**
  - Size of array.
- ▶ **length**
  - Length of vector.



# Matrix Operation

- ▶ Common errors in matrix operation

Command Window

```
>> A(100)  
Index exceeds matrix dimensions.
```

Command Window

```
>> A+B  
Error using +  
Matrix dimensions must agree.
```

Command Window

```
>> A*B  
Error using *  
Inner matrix dimensions must agree.
```

Command Window

```
>> C=[A B]  
Error using horzcat  
Dimensions of matrices being  
concatenated are not consistent.
```

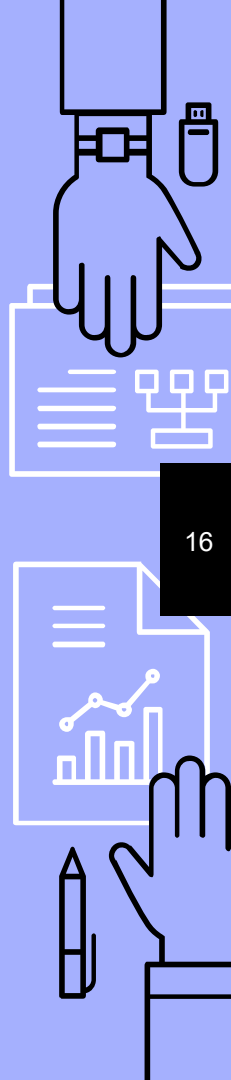
Command Window

```
>> C=[A; B]  
Error using vertcat  
Dimensions of matrices being  
concatenated are not consistent.
```



# Matrix Operation

- ▶ **A+B or A-B**
  - Matrix A and matrix B must have same matrix dimension.
  
- ▶ **A\*B**
  - Inner matrix dimension must agree.
  - Ex: A is  $M \times N$  and B is  $N \times P$





# Matrix Operation

- ▶  $A.*B$  or  $A./B$ 
  - Matrix A and matrix B must have same matrix dimension
  - The matrix operation was conducted element-by-element.

A =				
6	6	6	6	6
6	6	6	6	6
6	6	6	6	6
6	6	6	6	6
6	6	6	6	6

B =				
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2
2	2	2	2	2

```
>> A.*B  
  
ans =  
  
12 12 12 12 12  
12 12 12 12 12  
12 12 12 12 12  
12 12 12 12 12  
12 12 12 12 12
```

# Matrix Operation

- ▶  $A^k$ 
  - Matrix  $A$  must be a square matrix.
- ▶  $A.^k$ 
  - No specific requirement for matrix dimension.

```
A =  
  
 1  4  7  
 2  5  8  
 3  6  9
```

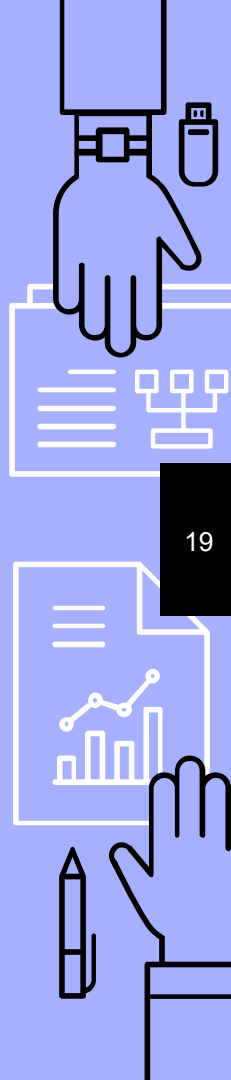
```
>> A^2  
  
ans =  
  
 30  66 102  
 36  81 126  
 42  96 150
```

```
>> A.^2  
  
ans =  
  
 1  16  49  
 4  25  64  
 9  36  81
```



# Matrix Operation

- ▶ **A/B**
  - **$A \cdot \text{inv}(B)$**
  - Inner matrix dimensions of A and inverse matrix of B must agree.
  
- ▶ **A\B**
  - **$\text{Inv}(A) \cdot B$**
  - Inner matrix dimensions of inverse matrix of A and B must agree.



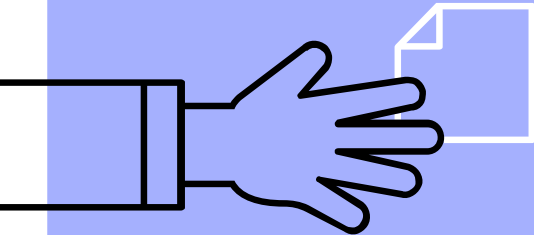
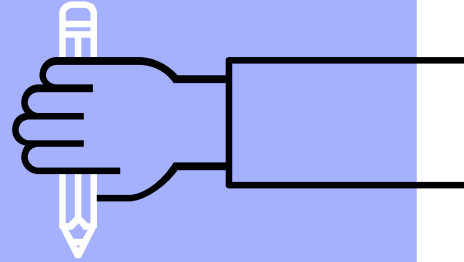
# Matrix Operation - Transpose

▶  $A'$

```
>> A  
  
A =  
  
    1    4    7  
    2    5    8  
    3    6    9
```

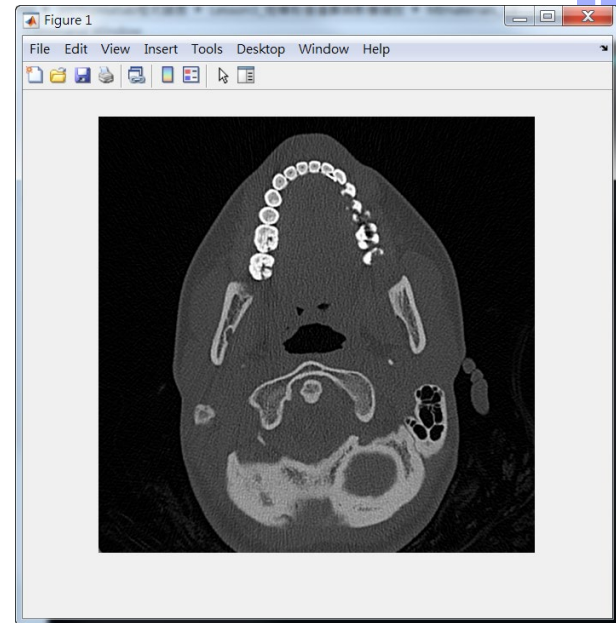
```
>> A'  
  
ans =  
  
    1    2    3  
    4    5    6  
    7    8    9
```

# Local Image Process and Resolution adjustment



# Step 1 – Import DICOM and Display

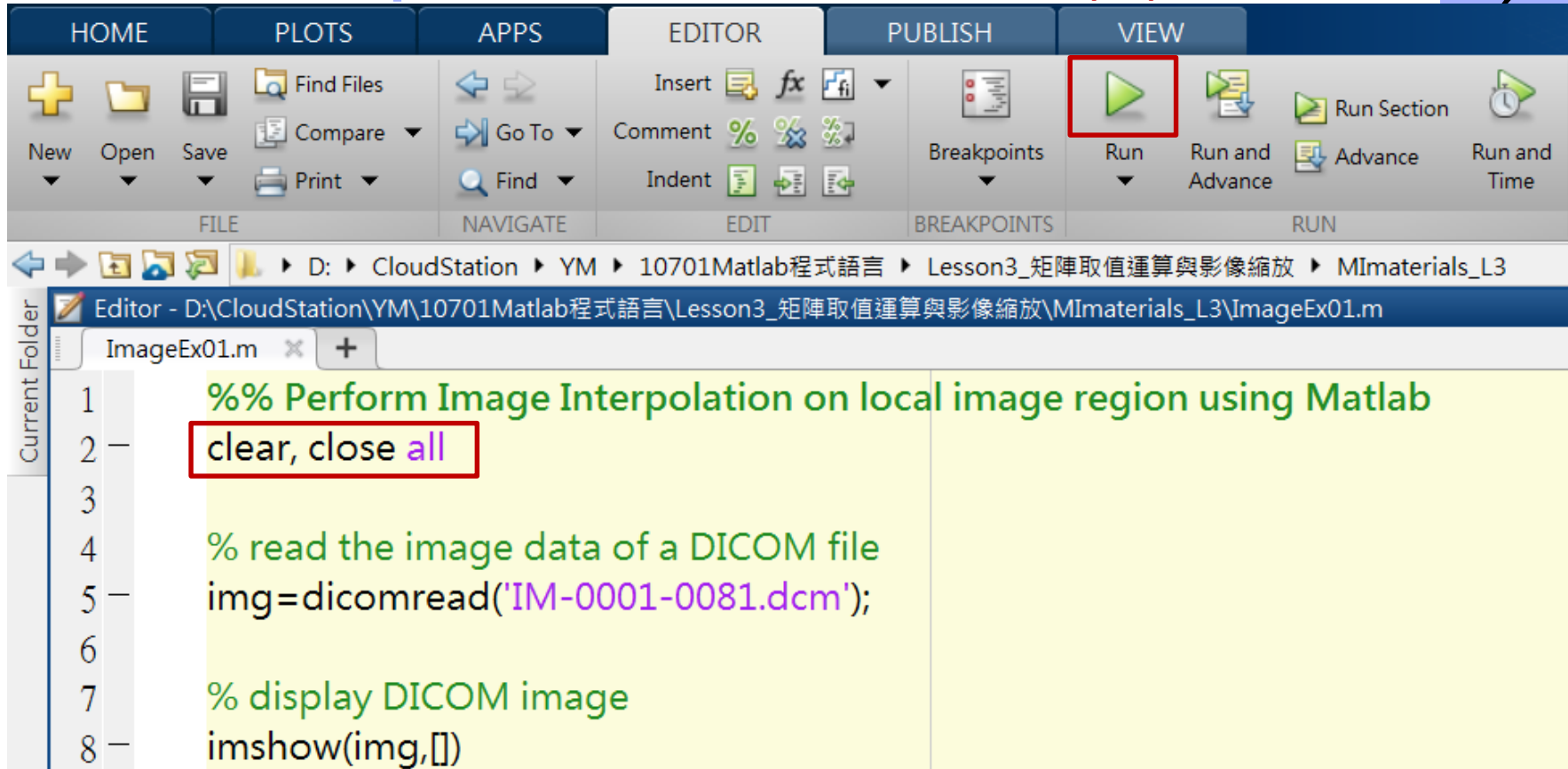
```
% get the image data of a DICOM file  
img=dicomread('IM-0001-0081.dcm');  
  
% display DICOM image  
imshow(img,[])
```



Please save as a Script File (\*.m)~

# Run a Script

(F5)

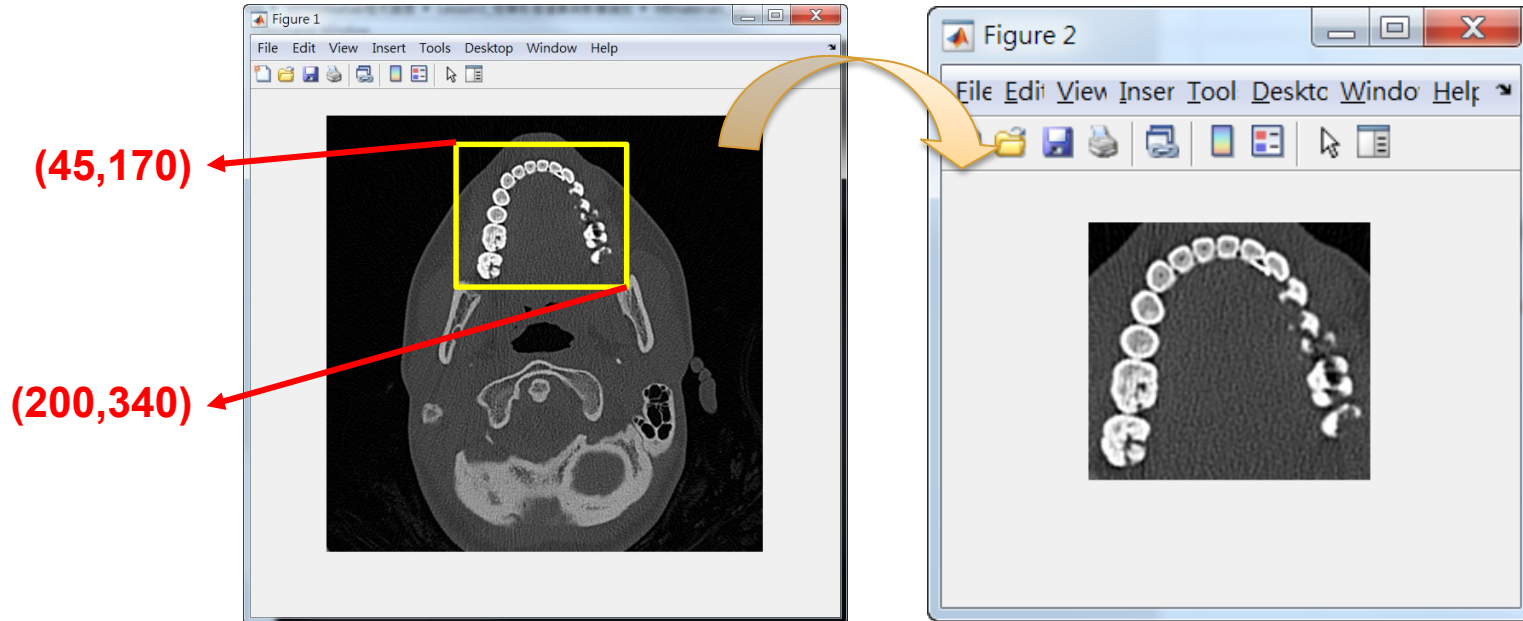


The screenshot shows the MATLAB software interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The VIEW tab is active, and the Run button (a green play icon) is highlighted with a red box. Below the menu bar, the toolbar contains various icons for file operations, navigation, editing, breakpoints, and running. The current workspace shows a script named 'ImageEx01.m' with the following code:

```
1 %% Perform Image Interpolation on local image region using Matlab  
2 clear, close all  
3  
4 % read the image data of a DICOM file  
5 img=dicomread('IM-0001-0081.dcm');  
6  
7 % display DICOM image  
8 imshow(img,[])
```

## Step 2 – Extract a Local Region

- ▶ Assign the yellow region to a new variable `img_local`
- ▶ Display `img_local` in a new figure





## Step 2 – Extract a Local Region

```
1 %% Perform Image Interpolation on local image region using Matlab
2 clear, close all
3
4 % get the image data of a DICOM file
5 img=dicomread('IM-0001-0081.dcm');
6
7 % display image
8 imshow(img,[])
9
10 % extract a local region and display in a new figure
11 img_local=img(45:200,170:340);
12 figure, imshow(img_local,[])
```

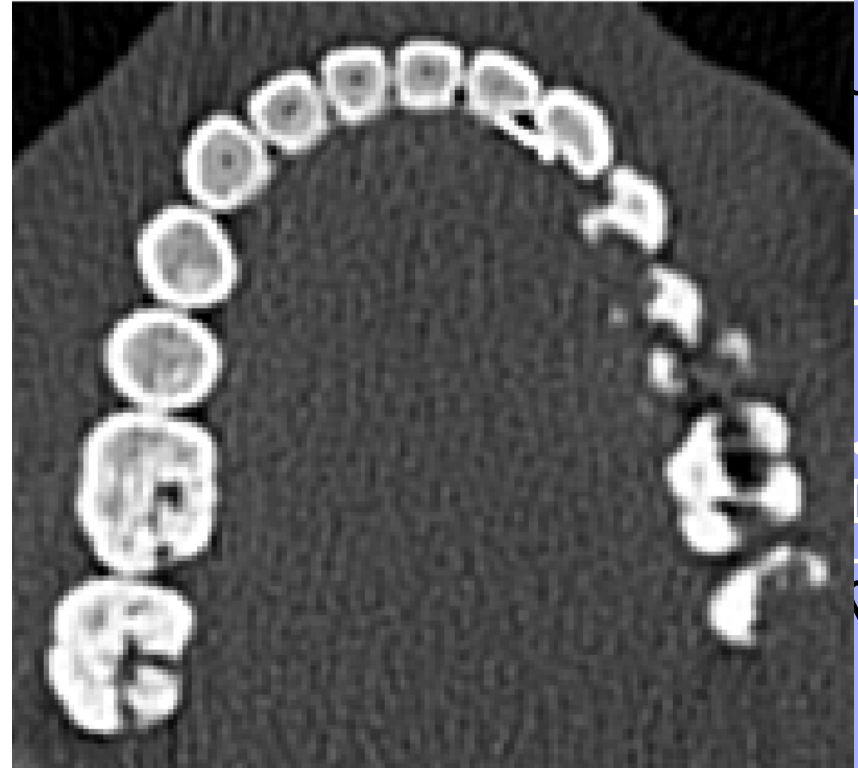
## Step 2 – Extract a Local Region

```
1 %% Perform Image Interpolation on local image region using Matlab
2 clear, close all
3
4 % get the image data of a DICOM file
5 img=dicomread('IM-0001-0081.dcm');
6
7 % display image
8 imshow(img,[])
9
10 % extract a local region and display in a new figure
11 img_local=img(45:200,170:340);
12 figure, imshow(img_local,[],'border','tight')
```

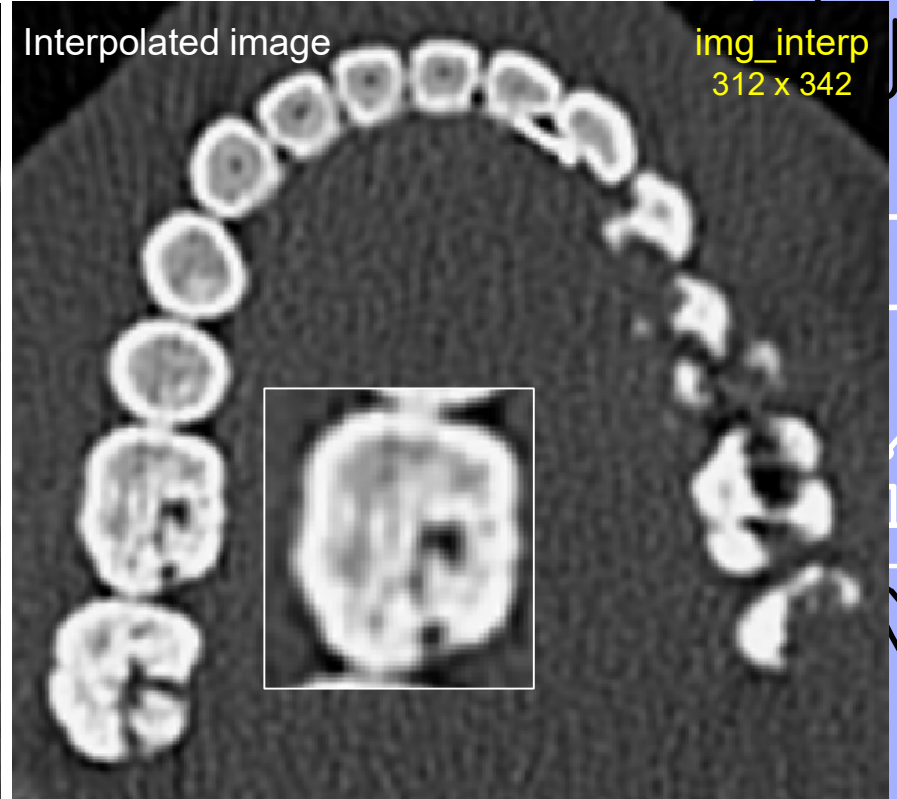
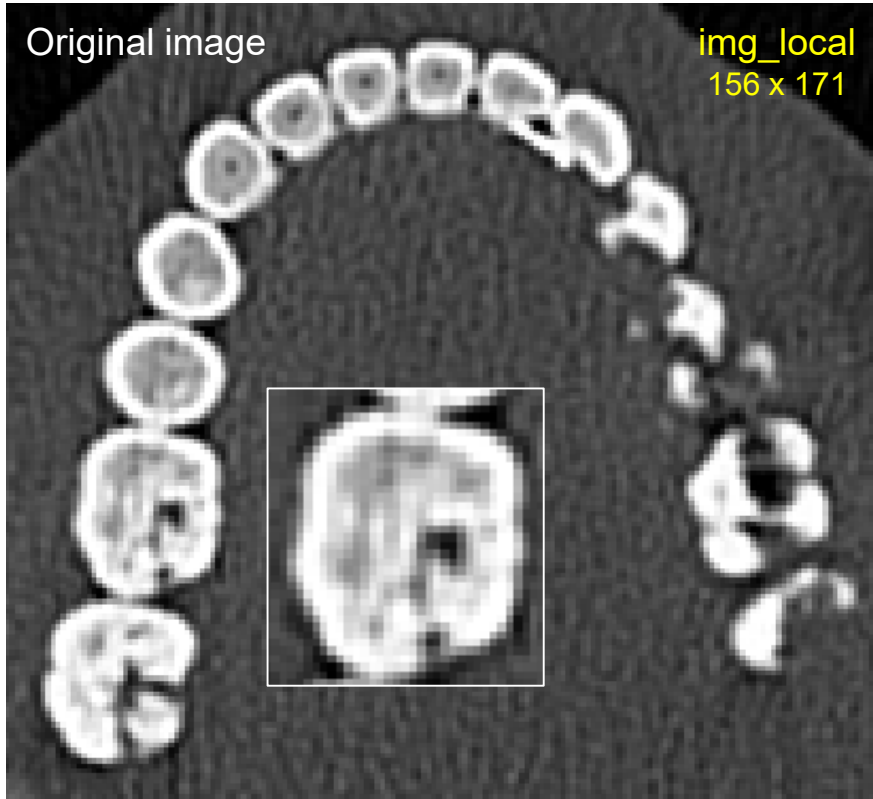
## Step 3 – Improve Resolution

- ▶ Low spatial resolution ⇔ Insufficient number of pixels
- ▶ Use **imresize** to increase the image size by a factor of 2.  
→ **image interpolation**
- ▶ Display interpolated image in a new figure

>> help **imresize**



# Step 3 – Improve Resolution



## Step 3 – Improve Resolution

```
4 % get the image data of a DICOM file
5 img=dicomread('IM-0001-0081.dcm');
6
7 % display image
8 imshow(img,[])
9
10 % extract a local region and display in a new figure
11 img_local=img(45:200,170:340);
12 figure, imshow(img_local,[],'border','tight')
13
14 % perform image interpolation
15 img_interp=imresize(img_local,2);
16 figure, imshow(img_interp,[],'border','tight')
```

# Homework

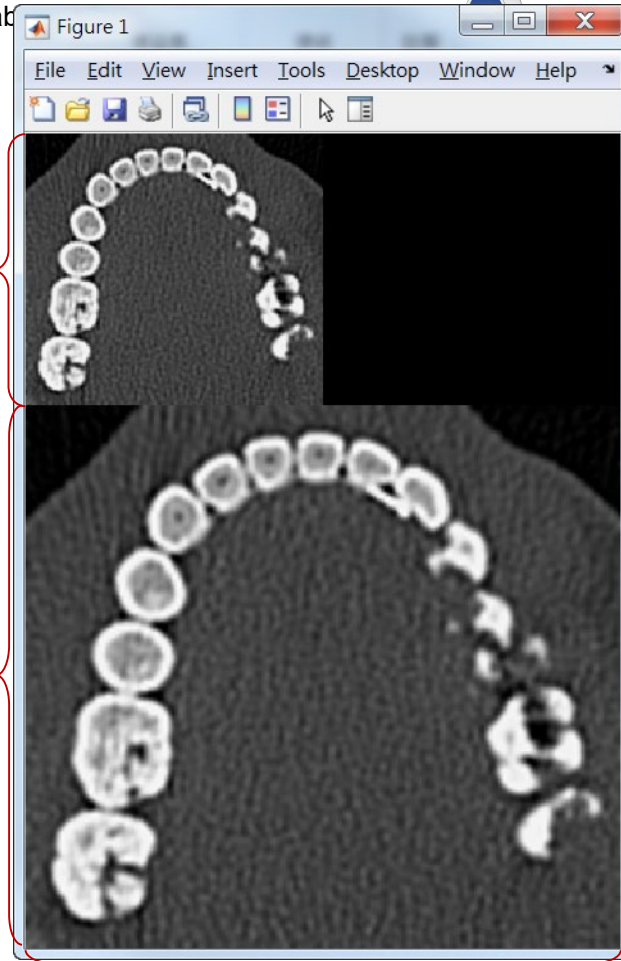
- ▶ Use **zeros** to create a new variable, **img\_comb**.
- ▶ Put both original and interpolated images into **img\_comb**.
- ▶ Display it in a new figure.

<http://cflu.lab>

156

312

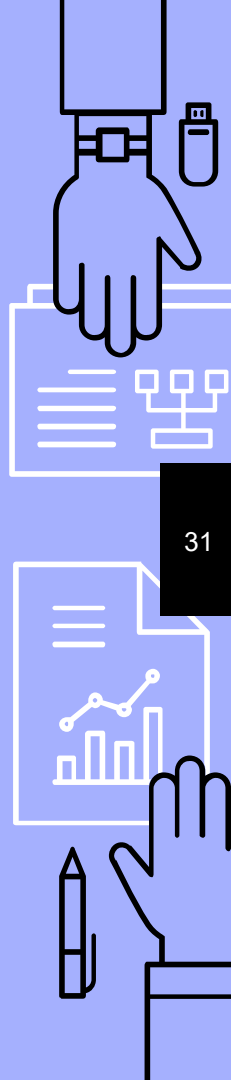
342

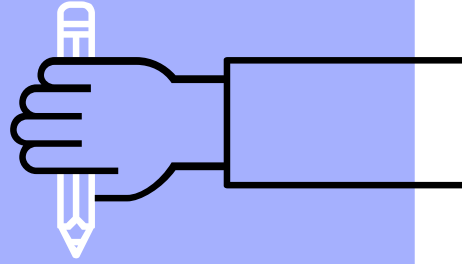




# Matrix generator **help** it!

- ▶ **ones**
  - Create a Ones array
- ▶ **zeros**
  - Create aZeros array
- ▶ **rand**
  - Create an array with uniformlydistributed pseudorandom numbers.
- ▶ **randn**
  - Create an array withnormally distributed pseudorandom numbers





THE END

[alvin4016@nycu.edu.tw](mailto:alvin4016@nycu.edu.tw)

