

Prof Associate ,  
School of information engineering Jiangxi university of  
science and technology, China

EMAIL: [ajm@jxust.edu.cn](mailto:ajm@jxust.edu.cn)

# Digital Image Processing

## 数字图像处理



Lecture 010:  
Using MATLAB Image-processing  
tool-box

Dr Ata Jahangir Moshayedi

Autumn \_2021



江西理工大学 信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



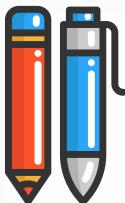
# Using MATLAB Image-processing tool-box

**LECTURE 010:**

SECTION 1: intro and basic

---

**Using MATLAB Image-processing tool-box**



# MATLAB = MATrix LABoratory

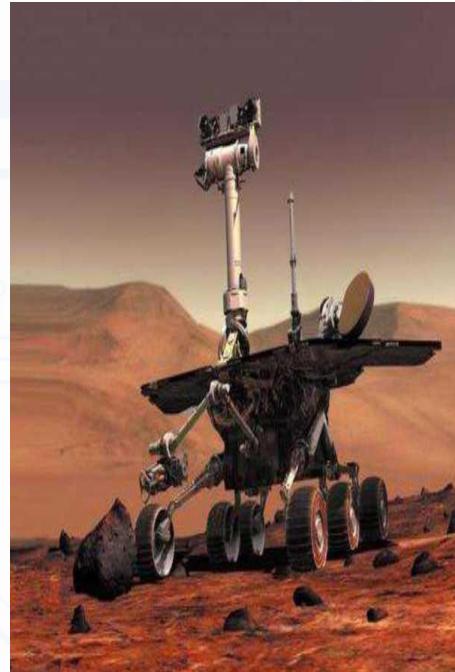


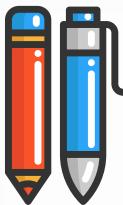
高级语言

- “MATLAB is a **high-level language** and interactive environment that enables us to perform computationally intensive tasks faster than with traditional programming languages such as C, C++ and Fortran.”

交互式解释语言

MATLAB = MATrix 实验室





# What is the Image Processing Toolbox?

什么是图像处理工具箱？



The Image Processing Toolbox is a collection of functions that extend the capabilities of the MATLAB's numeric computing environment.

图像处理工具箱是一组扩展MATLAB数值计算环境功能的函数。

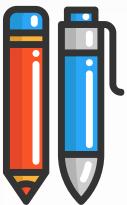
The toolbox supports a wide range of image processing operations, including:

工具箱支持多种图像处理操作，包括：

- Geometric operations
- Linear filtering and filter design
- Transforms
- Image analysis and enhancement
- Binary image operations
- Region of interest operations

- 几何运算
- 线性过滤和滤波器设计
- 变换- 图像分析和增强
- 二值图像操作
- 感兴趣的区域操作





# What Is the Image Processing Toolbox?



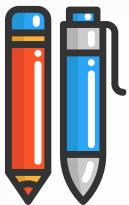
The Image Processing Toolbox is a collection of functions that extend the capability of the MATLAB® numeric computing environment. The toolbox supports a wide range of image processing operations, including:

- Spatial image transformations
- Morphological operations
- Neighborhood and block operations
- Linear filtering and filter design
- Transforms
- Image analysis and enhancement
- Image registration
- Deblurring
- Region of interest operations

- 空间图像变换
- 形态学操作
- 邻域和块操作
- 线性滤波和滤波器设计
- 变换
- 图像分析和增强
- 图像配准
- 去模糊
- 感兴趣的区域操作

Many of the toolbox functions are MATLAB M-files, a series of MATLAB statements that implement specialized image processing algorithms. You can view the MATLAB code for these functions using the statement





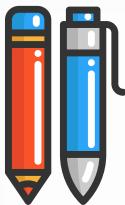
# What Is the Image Processing Toolbox?



图像处理工具箱是一组扩展MATLAB®数值计算环境功能的函数。工具箱支持多种图像处理操作，包括：

- 空间图像转换
- 形态操作
- 邻里和街区运营
- 线性滤波和滤波器设计
- 转变
- 图像分析和增强
- 图像注册
- 消除模糊
- 感兴趣区域业务

许多工具箱函数是MATLAB M-files，这是一系列实现专门图像处理算法的MATLAB语句。您可以使用以下语句查看这些函数的MATLAB代码



# Images in MATLAB



- MATLAB can import/export several image formats:

- BMP (Microsoft Windows Bitmap)
- GIF (Graphics Interchange Files)
- HDF (Hierarchical Data Format)
- JPEG (Joint Photographic Experts Group)
- PCX (Paintbrush)
- PNG (Portable Network Graphics)
- TIFF (Tagged Image File Format)
- XWD (X Window Dump)
- raw-data and other types of image data

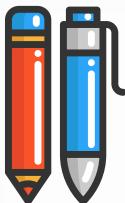
**Typically switch images to double to perform any processing and convert back to unsigned integer**

BMP (微软视窗位图)  
GIF (图形交换文件)  
HDF (分层数据格式)  
JPEG (联合图像专家组)  
PCX (画笔)  
PNG (便携式网络图形)  
TIFF (标记图像文件格式)  
XWD (X 窗口转储)  
原始数据和其他类型的图像数据

## MATLAB中的图像

- Data types in MATLAB
  - Double (64-bit double-precision floating point)
  - Single (32-bit single-precision floating point)
  - Int32 (32-bit signed integer)
  - Int16 (16-bit signed integer)
  - Int8 (8-bit signed integer)
  - Uint32 (32-bit unsigned integer)
  - Uint16 (16-bit unsigned integer)
  - Uint8 (8-bit unsigned integer)

MATLAB 中的数据类型双精度  
(64 位双精度浮点)  
单 (32 位单精度浮点)  
Int32 (32 位有符号整数)  
Int16 (16 位有符号整数)  
Int8 (8 位有符号整数)  
Uint32 (32 位无符号整数)  
Uint16 (16 位无符号整数)  
Uint8 (8 位无符号整数)



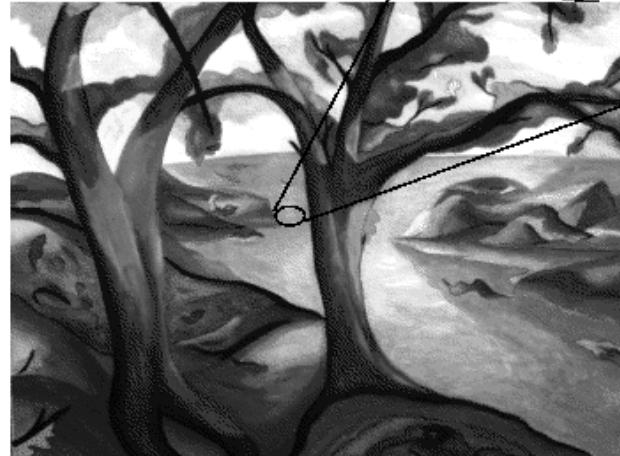
# Images in MATLAB



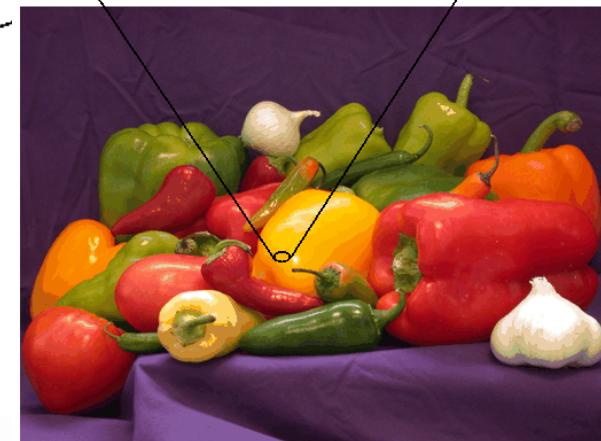
- Binary images : {0,1}
- Intensity images : [0,1] or uint8, double etc.
- RGB images :  $m \times n \times 3$
- Multidimensional images:  $m \times n \times p$  ( $p$  is the number of layers)

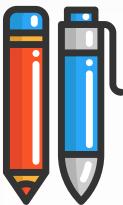
二进制图像：

- 强度图像：
- RGB 图像：
- 多维图像：



Blue	Green	Red	Blue	Green	Red
0.2251	0.2563	0.2235	0.2902	0.0627	0.2902
0.5342	0.2051	0.2157	0.5804	0.1922	0.2902
0.5342	0.1789	0.1307	0.5176	0.1922	0.2902
0.4308	0.2483	0.2624	0.5176	0.1294	0.1294
3344	0.2624	0.5176	0.1294	0.2588	0.2588
		0.1608	0.0627	0.2235	0.2588
		0.2235	0.5490	0.7765	0.7765
		0.5176	0.1608	0.2588	0.2588
		0.3882	0.5176	0.5804	0.5804
		0.2588	0.2902	0.2588	0.2588
		0.2235	0.1608	0.2588	0.2588
		0.2588	0.1608	0.2588	0.2588

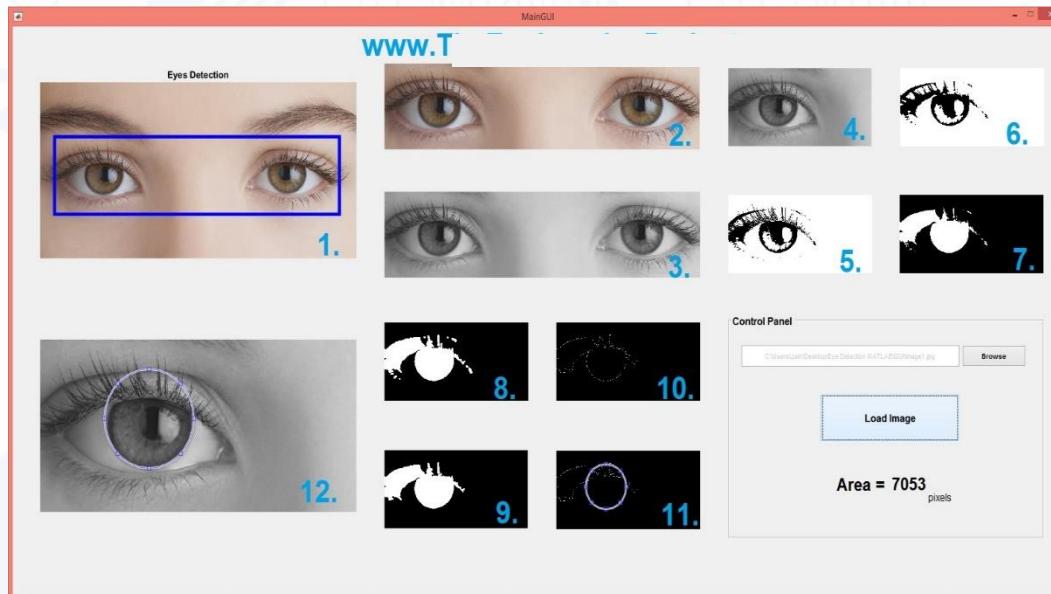




# MATLAB



- You can extend the capabilities of the Image Processing Toolbox by writing your own M-files, or by using the toolbox in combination with other toolboxes, such as the Signal Processing Toolbox and the Wavelet Toolbox.
- **Clear the MATLAB workspace of any variables and close open figure windows.**
- **clear, close all**

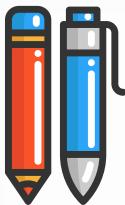


您可以通过编写自己的M文件，或将图像处理工具箱与其他工具箱（如信号处理工具箱和小波工具箱）结合使用，来扩展图像处理工具箱的功能。

清除MATLAB工作区中的所有变量，并关闭打开的图形窗口。

清除，全部关闭





# Read and Display an Image



读取和显示图像

dú qǔ hé xiǎn shì tú xiàng

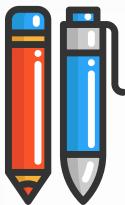
- To read an image, use the `imread` command.
- Let's read in a TIFF image named `pout.tif` (which is one of the sample images that is supplied with the Image Processing Toolbox), and store it in an array named `I`.

`I=imread('pout.tif');`

- Now call `imshow` to display `I`.

`imshow(I)`





# Read and Display an Image

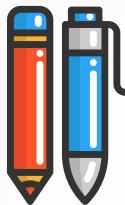


读取和显示图像

dú qǔ hé xiǎn shì tú xiàng

- 要读取图像，请使用 `imread` 命令。
- 让我们读入名为 `pout.tif` 的 TIFF 图像（这是图像处理工具箱提供的示例图像之一），并将 其存储在名为 `i` 的数组中。
- `I=imread ('pout.tif');`
- 现在调用 `imshow` 来显示 `i`。
- `imshow (I)`





# Some image source in matlab

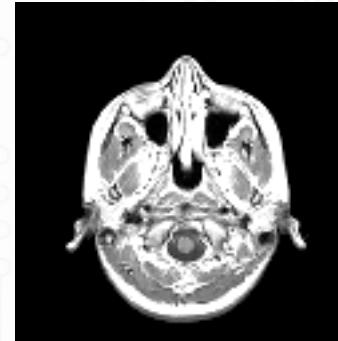
matlab中的一些图像源

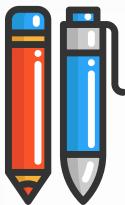


C:\Program Files\MATLAB\R2018a\toolbox\images\imdata

AT3\_1m4\_01.tif  
AT3\_1m4\_03.tif  
AT3\_1m4\_05.tif  
AT3\_1m4\_07.tif  
AT3\_1m4\_09.tif  
autumn.tif  
blobs.png  
cameraman.tif  
cell.tif  
circles.png  
coins.png  
concordorthophoto.png  
fabric.png  
forest.tif  
glass.png  
hestain.png  
liftingbody.png  
m83.tif  
moon.tif  
office\_1.jpg  
office\_3.jpg  
office\_5.jpg  
onion.png  
pears.png  
pillsetc.png  
rice.png  
shadow.tif  
spine.tif  
testpat1.png  
tire.tif  
trees.tif  
westconcordorthophoto.png

AT3\_1m4\_02.tif  
AT3\_1m4\_04.tif  
AT3\_1m4\_06.tif  
AT3\_1m4\_08.tif  
AT3\_1m4\_10.tif  
bag.png  
board.tif  
canoe.tif  
circbw.tif  
circuit.tif  
concordaerial.png  
eight.tif  
football.jpg  
gantrycrane.png  
greens.jpg  
kids.tif  
logo.tif  
mandi.tif  
mri.tif  
office\_2.jpg  
office\_4.jpg  
office\_6.jpg  
paper1.tif  
peppers.png  
pout.tif  
saturn.png  
snowflakes.png  
tape.png  
text.png  
tissue.png  
westconcordaerial.png





# Read and Display an Image



读取和显示图像

- If we want to read from different location

**A=imread('f:\ATA.jpg');**

- Now call `imshow` to display A.

**imshow(A)**

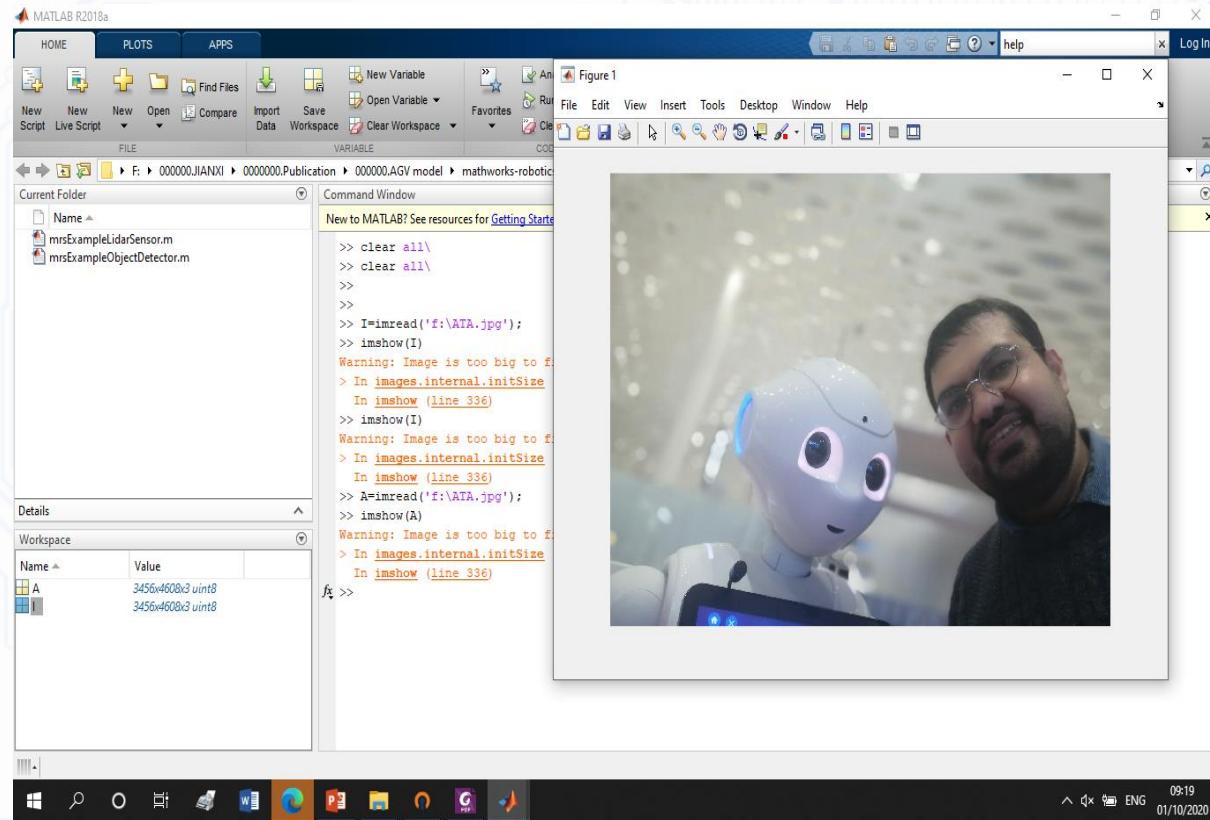
如果我们想从不同的位置阅读

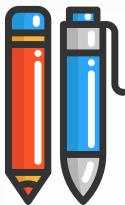
`A=imread`

`('f:\ATA.jpg')` ;

现在调用`imshow`以显示A。

`imshow (A)`





# Image Import and Export

图片导入导出



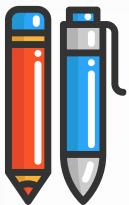
- Read and write images in Matlab 在Matlab中读写图像

```
img = imread('apple.jpg');  
dim = size(img);  
figure;  
imshow(img);  
imwrite(img, 'output.bmp', 'bmp');
```

- Alternatives to imshow imshow的替代方案

```
imagesc(I)  
imtool(I)  
image(I)
```





# Images and Matrices

图像和矩阵



## How to build a matrix(or image)?

如何构建矩阵（或图像）？

### Intensity Image:

row = 256;

col = 256;

img = zeros(row, col);

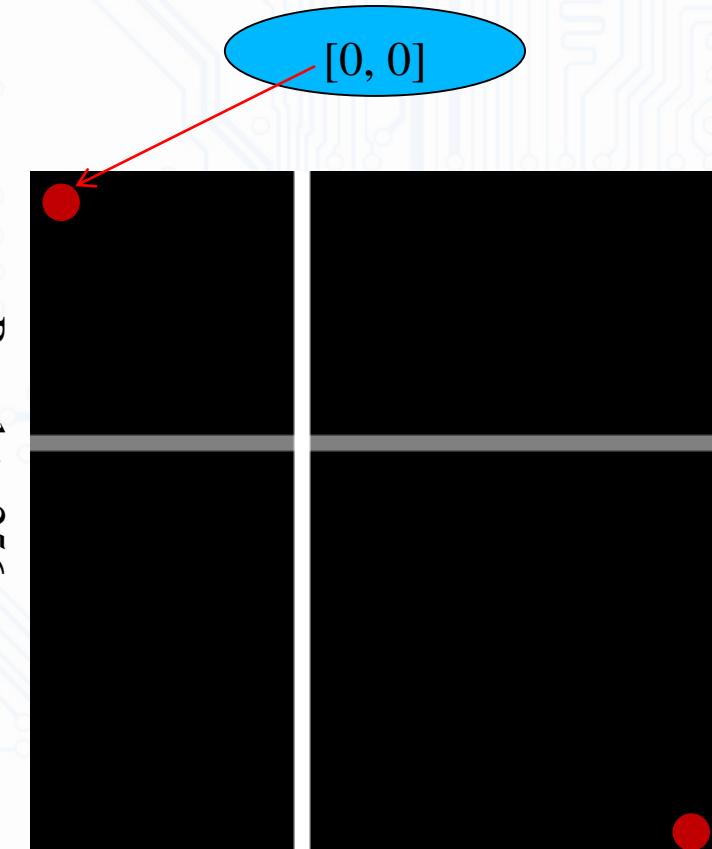
img(100:105, :) = 0.5;

img(:, 100:105) = 1;

figure;

imshow(img);

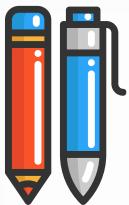
强度图像:



Column 1 to 256

[256, 256]





# Images and Matrices

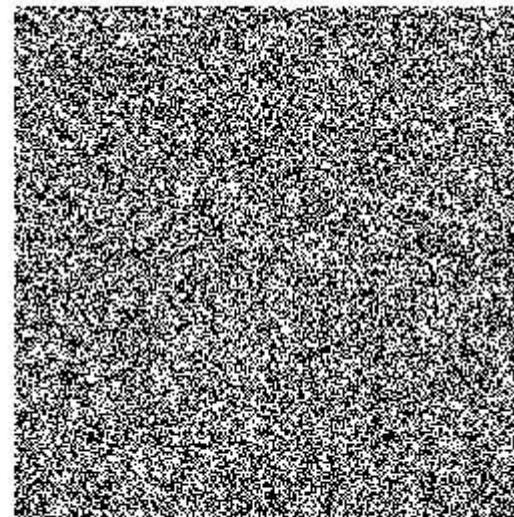


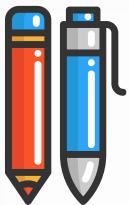
图像和矩阵

tú xiàng hé jǔ zhèn

**Binary Image:** 二进制图像:

```
row = 256;  
col = 256;  
img = rand(row, col)  
img = round(img);  
figure;  
imshow(img);  
size(im)
```





# Binary Images

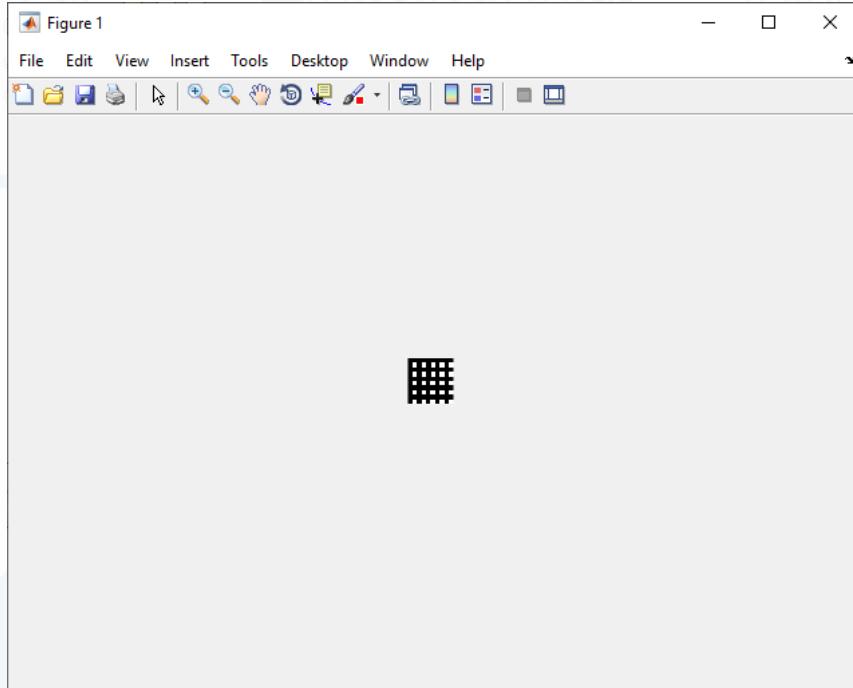
二进制图像



They are also called “ Black & White ”  
images containing ‘1’ for white and ‘0’ (zero)

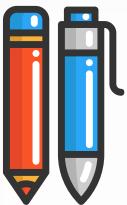
它们也被称为“黑白”

包含白色的“1”和“0”（零）  
的图像



```
clear  
x=zeros(10,10);  
x(2:2:10,2:2:10)=1;  
imshow(x)  
x(2:2:10,2:2:9)=1;  
imshow(x)
```





# Information about graphics file

有关图形文件的信息



File details including file name, data, size, format, height and width can be obtained using the command:

**imfinfo**

Information about graphics file

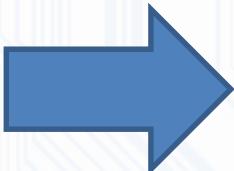
**Syntax**

info = imfinfo(filename)

info = imfinfo(filename,fmt)

info = imfinfo(URL)

info = imfinfo('f:\ATA.jpg')



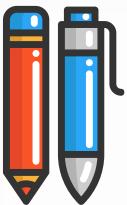
Information fields displayed by 'imfinfo' can be captured into a structural variable that can be used for subsequent computation.

info =

struct with fields:

```
Filename: 'f:\ATA.jpg'  
FileModDate: '27-Jan-2020 13:39:34'  
FileSize: 4933254  
Format: 'jpg'  
FormatVersion: ''  
Width: 4608  
Height: 3456  
BitDepth: 24  
ColorType: 'truecolor'  
FormatSignature: ''  
NumberOfSamples: 3  
CodingMethod: 'Huffman'  
CodingProcess: 'Sequential'  
Comment: {}  
BitsPerSample: [8 8 8]  
ImageDescription: 'rpt'  
Make: 'HUAWEI'  
Model: 'JKM-AL00'  
Orientation: 0  
XResolution: 72  
YResolution: 72  
ResolutionUnit: 'Inch'  
Software: 'JKM-AL00 9.1.0.221(C00E32R1P6)'  
DateTime: '2020:01:27 18:09:34'  
YCbCrPositioning: 'Centered'  
DigitalCamera: [1x1 struct]  
GPSInfo: [1x1 struct]  
DeviceSettingDescription: [105 112 112 0]  
ExifThumbnail: [1x1 struct]
```





# Information about graphics file

有关图形文件的信息



可使用以下命令获取文件详细信息，包括文件名、数据、大小、格式、高度和宽度：

imfinfo

Information about graphics file

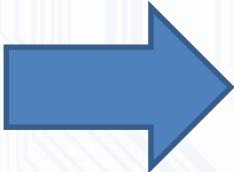
Syntax

info = imfinfo(filename)

info = imfinfo(filename,fmt)

info = imfinfo(URL)

info = imfinfo('f:\ATA.jpg')



“imfinfo”显示的信息字段可以捕获到结构变量中，用于后续计算

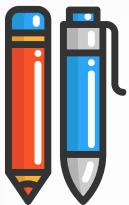
。

info =

struct with fields:

```
Filename: 'f:\ATA.jpg'  
FileModDate: '27-Jan-2020 13:39:34'  
FileSize: 4933254  
Format: 'jpg'  
FormatVersion: ''  
Width: 4608  
Height: 3456  
BitDepth: 24  
ColorType: 'truecolor'  
FormatSignature: ''  
NumberOfSamples: 3  
CodingMethod: 'Huffman'  
CodingProcess: 'Sequential'  
Comment: {}  
BitsPerSample: [8 8 8]  
ImageDescription: 'rpt'  
Make: 'HUAWEI'  
Model: 'JKM-AL00'  
Orientation: 0  
XResolution: 72  
YResolution: 72  
ResolutionUnit: 'Inch'  
Software: 'JKM-AL00 9.1.0.221(C00E32R1P6)'  
DateTime: '2020:01:27 18:09:34'  
YCbCrPositioning: 'Centered'  
DigitalCamera: [1x1 struct]  
GPSInfo: [1x1 struct]  
DeviceSettingDescription: [105 112 112 0]  
ExifThumbnail: [1x1 struct]
```





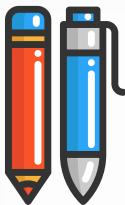
# Information about graphics file

## 有关图形文件的信息



Information about the graphics file, returned as a structure array. The set of fields in info depends on the individual file and its format. This table lists the nine fields that always appear, and describes their values.

Field Name	Description	Value
Filename	Name of the file or the internet URL specified. If the file is not in the current folder, the character vector contains the full path name of the file.	character vector
FileModDate	Date when the file was last modified.	character vector
FileSize	Size of the file, in bytes.	integer
Format	File format, as specified by <i>fmt</i> . For formats with more than one possible extension (for example, JPEG and TIFF files), imfinfo returns the first variant in the file format registry.	character vector
FormatVersion	File format version.	character vector or number
Width	Image width, in pixels.	integer
Height	Image height, in pixels.	integer
BitDepth	Number of bits per pixel.	integer
ColorType	Image type. ColorType includes, but is not limited to, 'truecolor' for a truecolor (RGB) image, 'grayscale' for a grayscale intensity image, or 'indexed' for an indexed image.	character vector



# Check the Image in Memory

检查内存中的图像

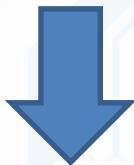


- Enter the `whos` command to see how `A` is stored in memory.

输入`whos`命令以查看如何在内存中存储。

`whos`

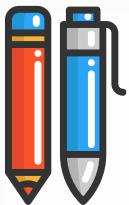
MATLAB responds with



`>> whos`

Name	Size	Bytes	Class	Attributes
A	3456x4608x3	47775744	uint8	
I	3456x4608x3	47775744	uint8	
info	1x1	17728	struct	





# Open Image Viewer app

打开图像查看器应用程序



**Imtool** : Open Image Viewer app

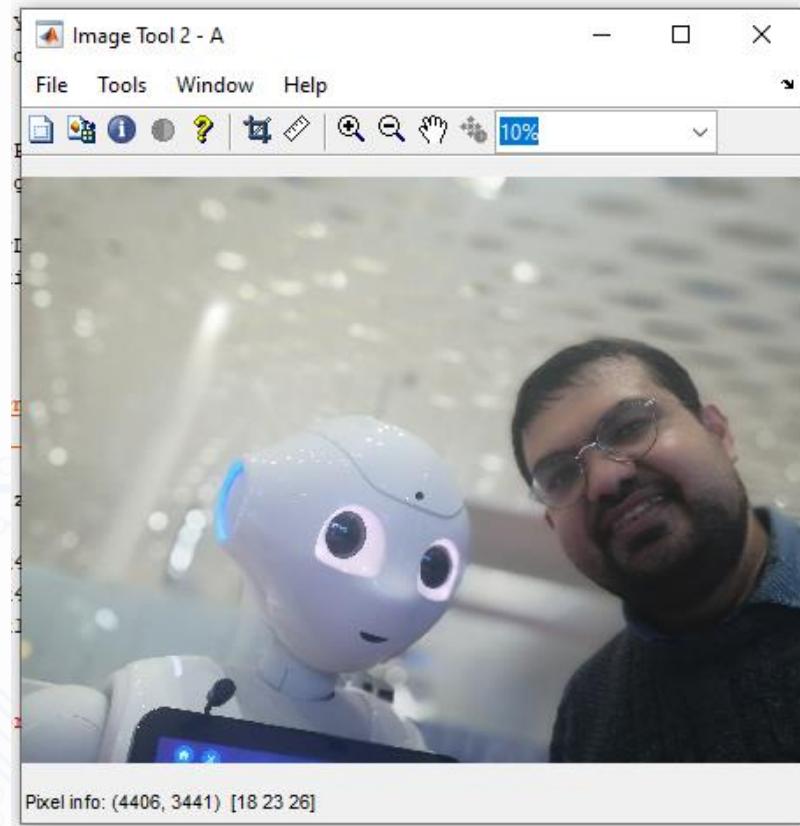
The Image Viewer presents an integrated environment for displaying images and performing common image processing tasks.

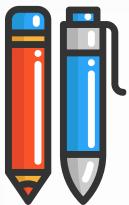
The Image Viewer provides all the image display capabilities of [imshow](#), which optimizes figure, axes, and image object property settings for image display. The Image Viewer also provides access to several tools for navigating and exploring images, such as the Pixel Region tool, Image Information tool, and the Adjust Contrast tool.

## Syntax

```
imtool  
imtool(I)  
imtool(I,range)  
imtool(RGB)  
imtool(BW)  
imtool(X,cmap)  
imtool(filename)  
imtool(___,Name,Value)  
hfigure = imtool(___)  
imtool close all
```

>> imtool(A)





# Open Image Viewer app

打开图像查看器应用程序



Imtool: 打开图像查看器应用程序

图像查看器提供了一个用于显示图像和执行常见图像处理任务的集成环境。

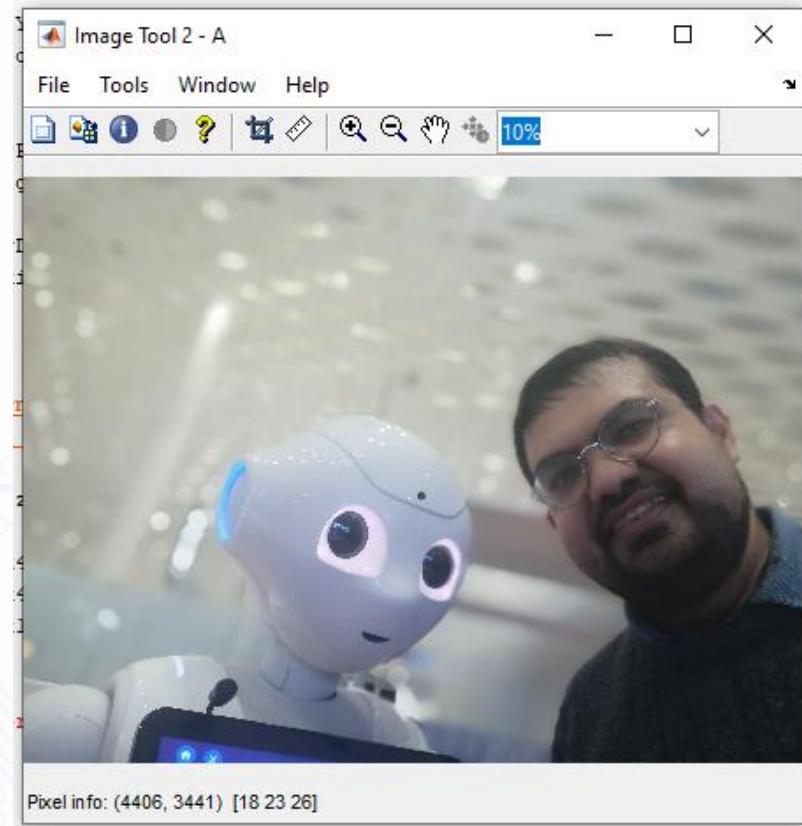
图像查看器提供imshow的所有图像显示功能，它优化图像显示的图形、轴和图像对象属性设置。图像查看器还提供了若干用于导航和浏览图像的工具，如像素区域工具、图像信息工具和调整对比度工具。

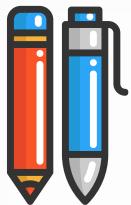
## 语法

### Syntax

```
imtool  
imtool(I)  
imtool(I,range)  
imtool(RGB)  
imtool(BW)  
imtool(X,cmap)  
imtool(filename)  
imtool(__,Name,Value)  
hfigure = imtool(__)  
imtool close all
```

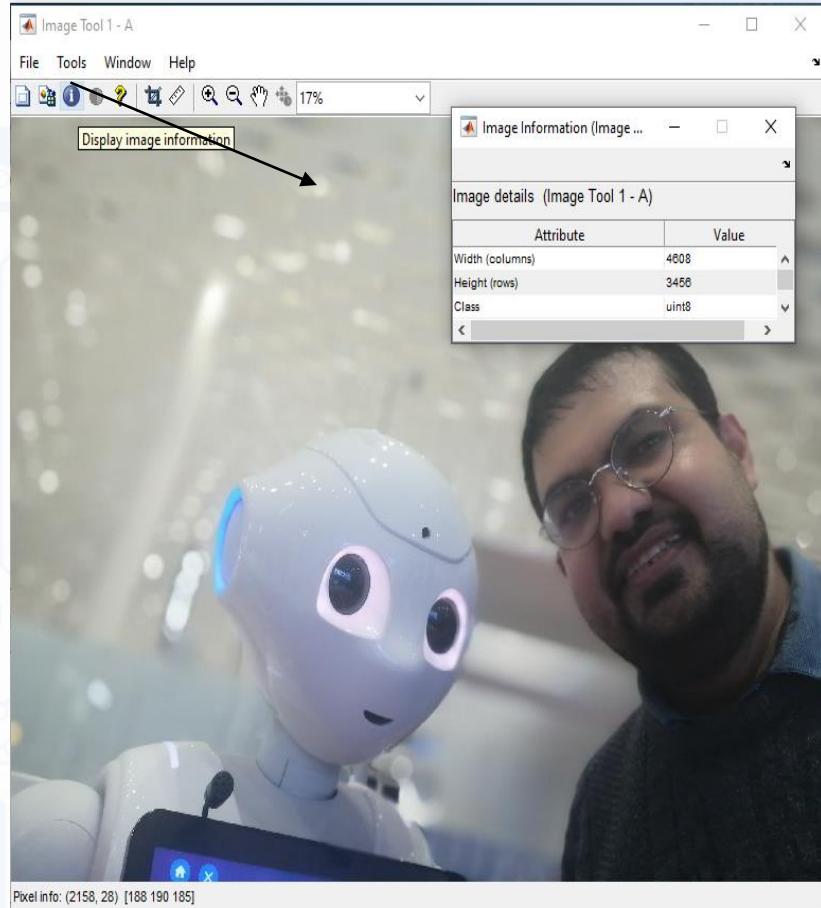
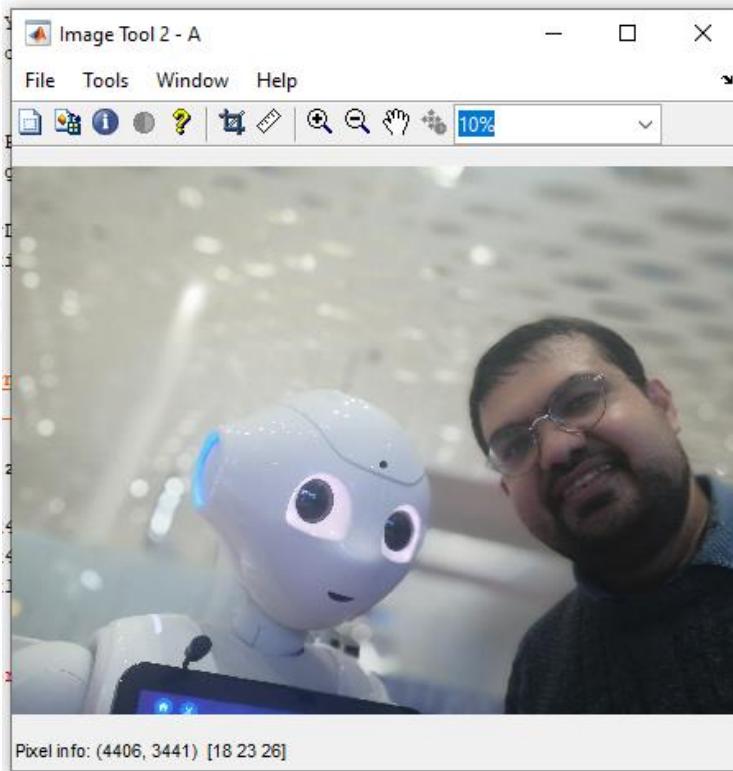
>> imtool(A)

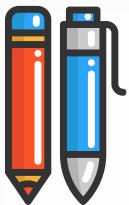




# Open Image Viewer app

打开图像查看器应用程序





# Measure Distance

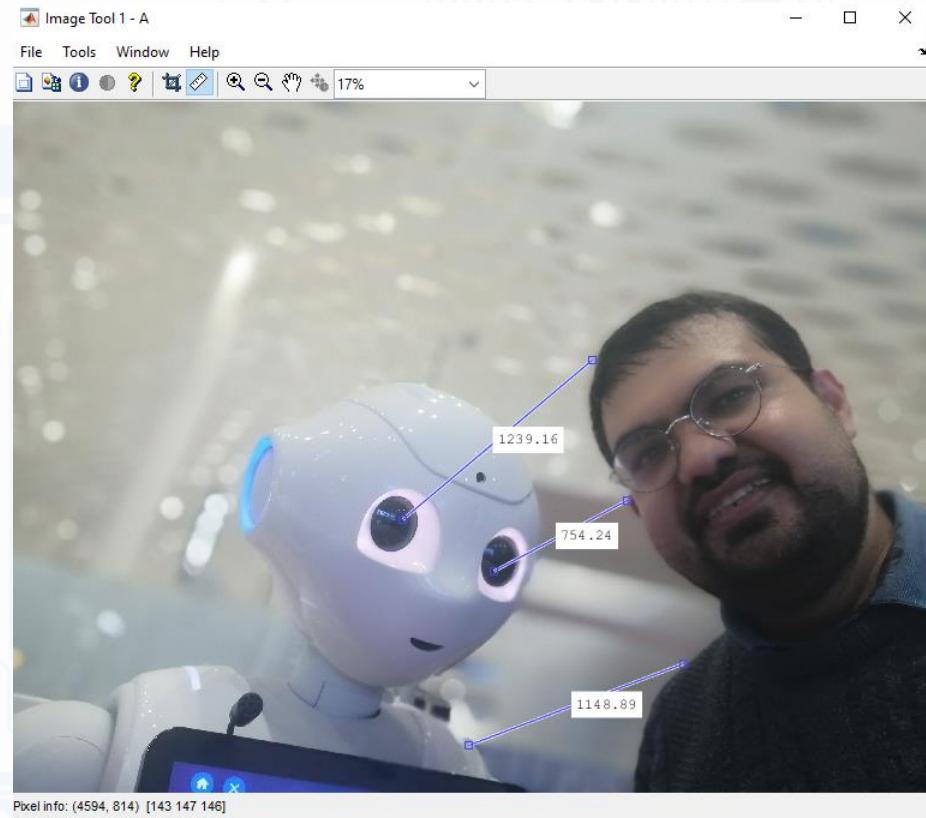
测量距离

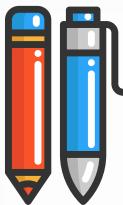
cè liàng jù lí



- The Measure Distance tool under Tools tab is used to show the distance between the two selected points. Fig shows the use of this tool to measure distances between the objects in figure

“工具”选项卡下的“测量距离”工具用于显示两个选定点之间的距离。图显示了使用此工具测量图中对象之间的距离



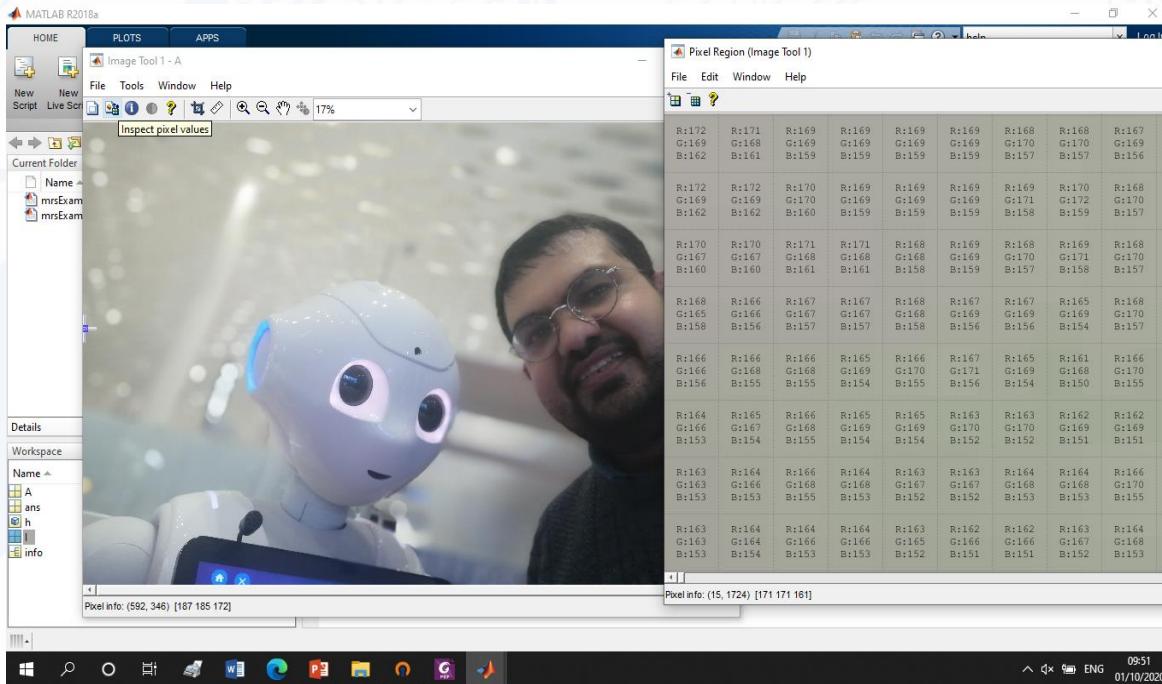


# Pixel Region tool

像素区域工具

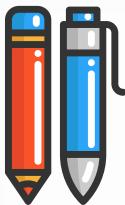


- The Overview tool under Tools tab shows the entire image in a thumbnail view. The Pixel Region tool shows individual pixels from the small square region on the upper right tip, zoomed large enough to see the actual pixel values. Fig. shows the snapshot.



“工具”选项卡下的“概述”工具在缩略图视图中显示整个图像。像素区域工具显示右上角小正方形区域中的单个像素，放大到足以看到实际像素值。图显示了快照。





# Multiple images

多张图片



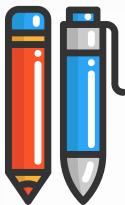
- Multiple images can be displayed within one figure using the subplot function.
- This function has three parameters within the brackets, where the first two parameters specify the number of rows and columns to divide the figure.
- The third parameter specifies which subdivision to use. For example, subplot(3,2,3) tells MATLAB to divide the figure into three rows and two columns and set the third cell as active.

使用子图功能可以在一个图形中显示多个图像。

此函数在括号内有三个参数，前两个参数指定要分割图形的行数和列数。

第三个参数指定要使用的细分。例如，子图（3,2,3）告诉MATLAB将图形分成三行两列，并将第三个单元格设置为活动。





# Multiple images

多张图片



clc

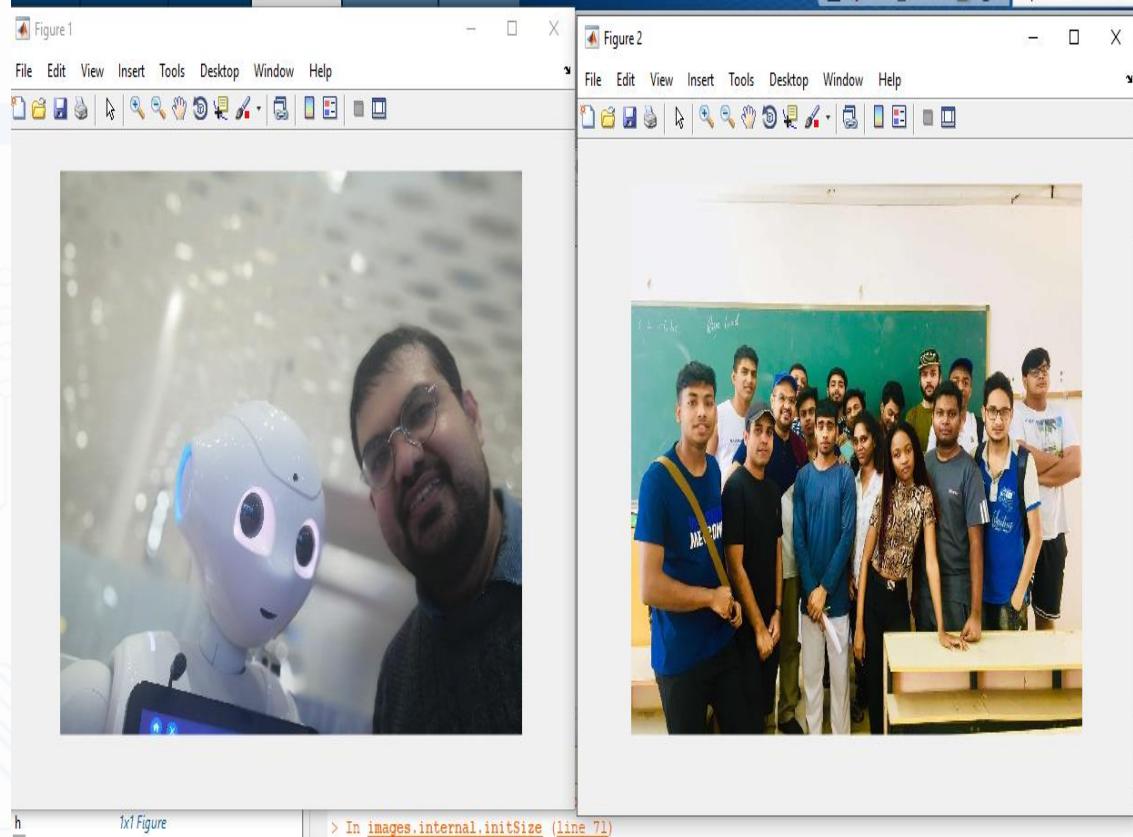
close all

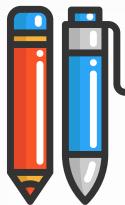
```
A=imread('f:\ATA.jpg');
```

```
figure();imshow(A)
```

```
B=imread('f:\ATA1.jpg');
```

```
figure();imshow(B)
```



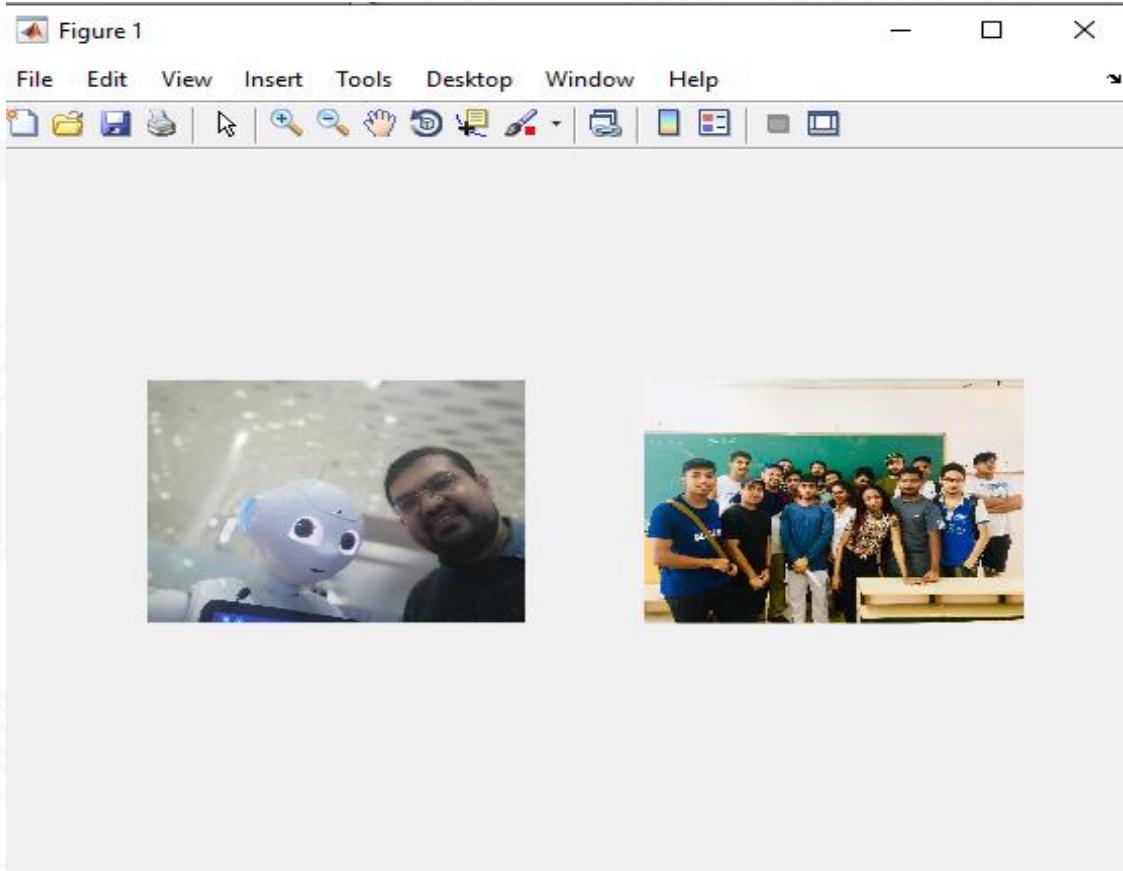


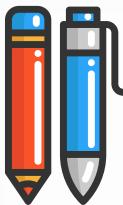
# Multiple images

多张图片



```
clc  
close all  
A=imread('f:\ATA.jpg');  
B=imread('f:\ATA1.jpg');  
figure  
subplot(1,2,1),imshow(A)  
subplot(1,2,2),imshow(B)
```





# Writing images

写图像

xiě tú xiàng



- Images are written to the current directory using the `imwrite` function, which has the following syntax:

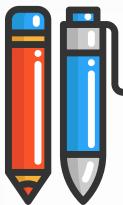
## Syntax

```
imwrite(A,filename)  
imwrite(A,map,filename)  
imwrite(__,fmt)  
imwrite(__,Name,Value)
```

*imwrite(f, 'filename');*

This command writes image data ‘f’ to the file specified by ‘filename’ in your current folder. The `imwrite` function supports most of the popular graphic file formats including GIF, HDF, JPEG or JPG, PBM, BMP, PGM, PNG, PNM, PPM and TIFF and so on.





# Writing images

写图像

xiě tú xiàng



- 使用imwrite函数将图像写入当前目录，该函数具有以下语法：

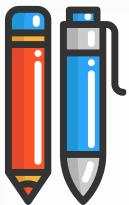
## Syntax

```
imwrite(A,filename)  
imwrite(A,map,filename)  
imwrite(__,fmt)  
imwrite(__,Name,Value)
```

*imwrite (f, “文件名”);*

此命令将图像数据“*f*”写入当前文件夹中由“*filename*”指定的文件。*imwrite*功能支持大多数流行的图形文件格式，包括*GIF*、*HDF*、*JPEG*或*JPG*、*PBM*、*BMP*、*PGM*、*PNG*、*PNM*、*PPM*和*TIFF*等。





# imwrite....



- The following example writes a  $100 \times 100$  array of grayscale values to a PNG file named random.png in the current folder:

以下示例将 $100 \times 100$ 灰度值数组写入当前文件夹中名为random.PNG的PNG文件：

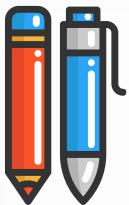
```
F = rand(100);  
imwrite(F, 'random.png')
```

When you open the folder in which you are working, an image file named ‘random.png’ is created.



打开正在工作的文件夹时，将创建一个名为“random.png”的图像文件。





# imwrite....



- For JPEG images, the imwrite syntax applicable is:

*imwrite(f, 'filename.jpg', 'quality', q)*

- where ‘q’ is an integer between 0 and 100.
- It can be used to reduce the image size. The quality parameter is used as a trade-off between the resulting image’s subjective quality and file size.

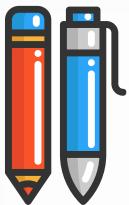
对于JPEG图像，适用的imwrite语法为：

`imwrite (f, 'filename.jpg', 'quality', q)`

其中“q”是介于0和100之间的整数。

它可以用来减小图像大小。质量参数用作结果图像的主观质量和文件大小之间的权衡。





# imwrite....

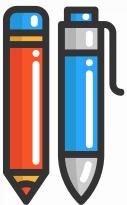


- You can now read the image ATA.jpg saved in the current working folder and save it in JPG format with three different quality parameters: 75 (default), 10 (poor quality and small size) and 90 (high quality and large size):

```
clc
close all
A=imread('f:\ATA.jpg');
B=imread('f:\ATA1.jpg');
imwrite(A, 'ATA.jpg', 'quality', 75);
figure(); imshow(A)
imwrite(A, 'ATA.jpg', 'quality', 10);
figure(); imshow(A)
imwrite(A, 'ATA.jpg', 'quality', 90);
figure(); imshow(A)
```

现在，您可以读取保存在当前工作文件夹中的图像ATA.jpg，并将其保存为jpg格式，其中包含三个不同的质量参数：75（默认）、10（质量差且尺寸小）和90（质量高且尺寸大）：





clc

close all

```
A=imread('f:\ATA.jpg');
```

```
B=imread('f:\ATA1.jpg');
```

```
imwrite(A,'ATA.jpg','quality',75);
```

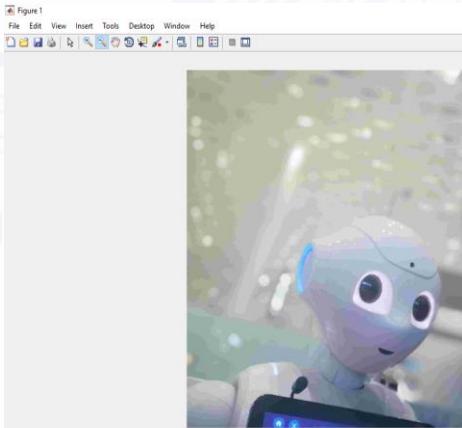
```
figure();imshow(A)
```

```
imwrite(A,'ATA.jpg','quality',10);
```

```
figure();imshow(A)
```

```
imwrite(A,'ATA.jpg','quality',90);
```

```
figure();imshow(A)
```



# imwrite....

Figure 3  
File Edit View Insert Tools Desktop Window Help

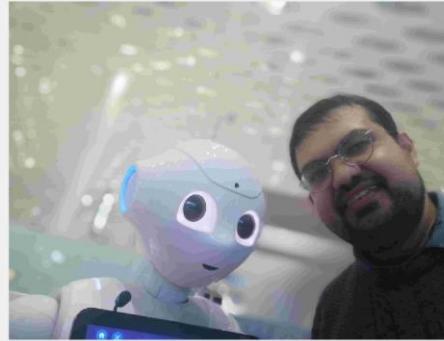


Figure 2  
File Edit View Insert Tools Desktop Window Help

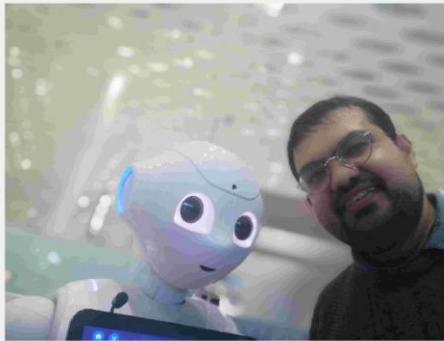
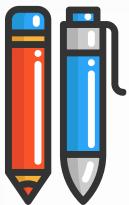


Figure 1  
File Edit View Insert Tools Desktop Window Help

Figs show images for quality factors of 75, 10 and 90, respectively. You will note that that a low-quality image has a smaller size than a higher-quality image.



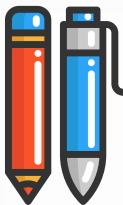


# imwrite....



- The imwrite syntax applicable to TIFF images has the format: 适用于TIFF图像的imwrite语法具有以下格式:
- *imwrite (g, 'filename.tif', 'compression', 'parameter', ..... 'resolution',[colresrowres])*





# Types of images

图像类型

tú xiàng lèi xíng



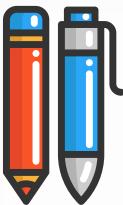
The MATLAB tool box supports four types of images, namely, grey-level images, binary images, indexed images and RGB images. Brief description of these image types is given below.

MATLAB工具箱支持四种类型的图像，即灰度图像、二值图像、索引图像和RGB图像。下面给出了这些图像类型的简要说明。

Most monochrome image processing algorithms are carried out using binary or greyscale images.

大多数单色图像处理算法都是使用二进制或灰度图像进行的。

Grey-level images	Also referred to as monochrome images, these use 8 bits per pixel, where a pixel value of 0 corresponds to ‘black,’ a pixel value of 255 corresponds to ‘white’ and intermediate values indicate varying shades of grey. These are also encoded as a 2D array of pixels, with each pixel having 8 bits.
Binary images	These images use 1 bit per pixel, where a 0 usually means ‘black’ and a 1 means ‘white.’ These are represented as a 2D array. Small size is the main advantage of binary images.
Indexed images	These images are a matrix of integers (X), where each integer refers to a particular row of RGB values in a secondary matrix (map) known as a colour map.
RGB image	In an RGB image, each colour pixel is represented as a triple containing the values of its R, G and B components. In MATLAB, an RGB colour image corresponds to a 3D array of dimensions $M \times N \times 3$ . Here ‘M’ and ‘N’ are the image’s height and width, respectively, and 3 is the number of colour components. For RGB images of class double, the range of values is [0.0, 1.0], and for classes uint8 and uint16, the ranges are [0, 255] and [0, 65535], respectively.



# Image quality

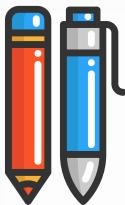
画面质量

huà miàn zhì liàng



- **Image quality is defined in terms of spatial resolution and quantisation.**
- Spatial resolution is the pixel density over the image. The greater the spatial resolution, the more are the pixels used to display the image. Spatial resolution is expressed qualitatively as dots per inch (dpi).
- The image resolution can be changed using the imresize function. The command `imresize(x,1/2)` halves the image size. This is done by taking a matrix from the original matrix having elements whose row and column indices are even. `imresize(x,2)` means all the pixels are repeated to produce an image of the same size as original, but with half the resolution in each direction.
- Other resolution changes can be done by using the desired scaling factors. Command `imresize(imresize(x,1/2),2)` changes the image resolution by half while keeping the image size same. Similarly, command `imresize(imresize(x,1/4),4)` changes the image resolution by one-fourth.





# Image quality

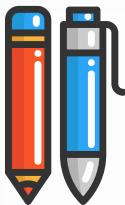
画面质量

huà miàn zhì liàng



- 图像质量定义为空间分辨率和量化。
- 空间分辨率是图像上的像素密度。空间分辨率越高，用于显示图像的像素越多。空间分辨率定性地表示为每英寸点数 (dpi) 。
- 可以使用imresize功能更改图像分辨率。命令imresize ( $x, 1/2$ ) 将图像大小减半。这是通过从原始矩阵中提取一个矩阵来实现的，该矩阵包含行和列索引为偶数的元素。imresize ( $x, 2$ ) 表示重复所有像素以生成与原始图像大小相同的图像，但每个方向的分辨率为原始图像的一半。
- 其他分辨率更改可以通过使用所需的比例因子来完成。命令imresize ( $imresize (x, 1/2), 2$ ) 将图像分辨率更改一半，同时保持图像大小不变。类似地，命令imresize ( $imresize (x, 1/4), 4$ ) 将图像分辨率更改四分之一。





# Image quality

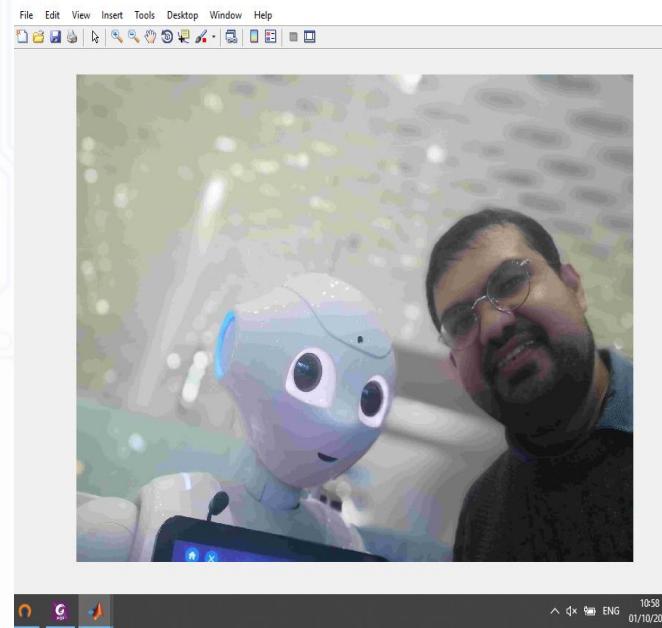
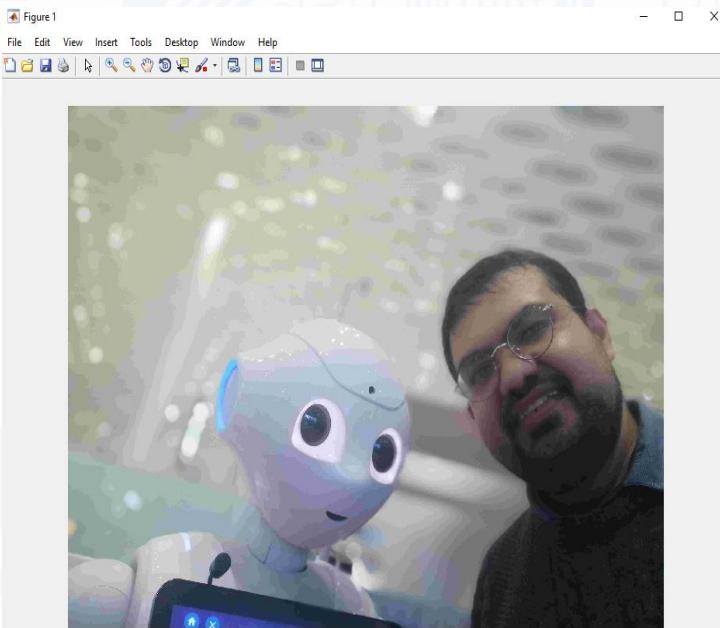
画面质量

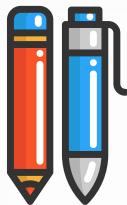


The following set of commands reduces the resolution of image ‘ATA.jpg’ by half:

```
clc  
close all  
A = imread('f:\ATA.jpg');  
figure();imshow(A)  
A1 = imresize((imresize(A,1/2)),2);  
figure();imshow(A1)
```

以下命令集将图像“ATA.jpg”的分辨率降低一半：



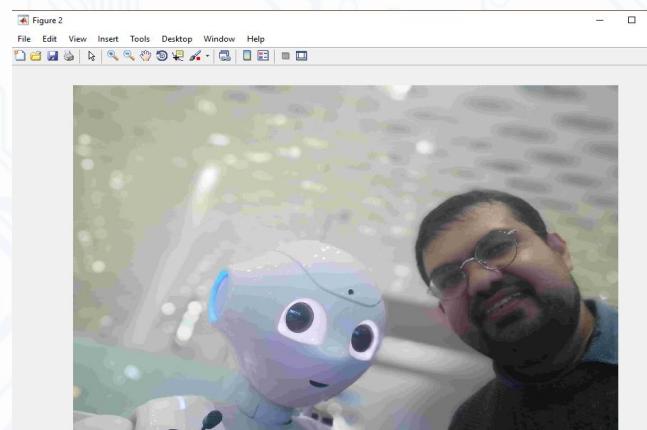
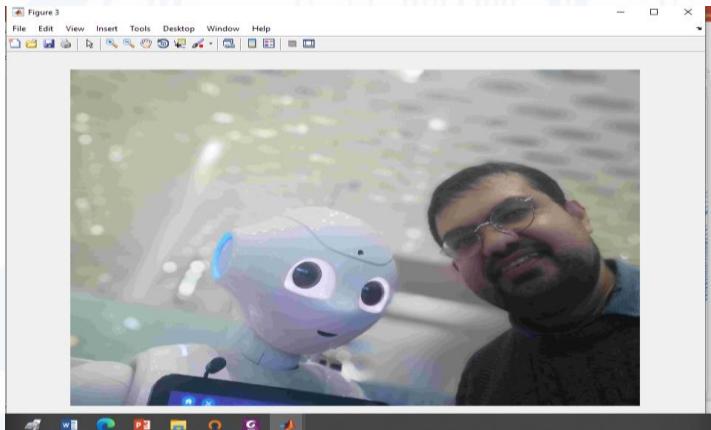
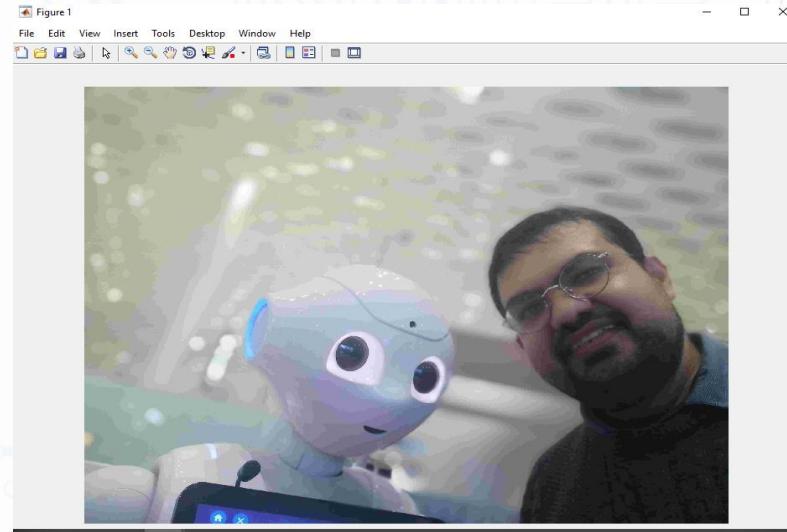


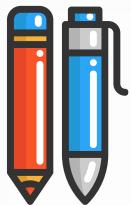
# Image quality

画面质量



```
clc  
close all  
A = imread('f:\ATA.jpg');  
figure();imshow(A)  
A1 = imresize((imresize(A,1/2)),2);  
figure();imshow(A1)  
A2 = imresize((imresize(A,1/4)),4);  
figure();imshow(A2)
```



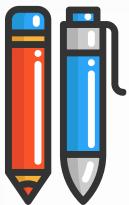


# YOU test

你测试

```
clc  
close all  
  
A = imread('f:\ATA.jpg');  
  
figure();imshow(A)  
  
A1 = imresize((imresize(A,1/2)),2);  
  
figure();imshow(A1)  
  
A2 = imresize((imresize(A,1/4)),4);  
  
figure();imshow(A2)  
  
A3 = imresize((imresize(A,1/8)),8);  
  
figure();imshow(A3)
```



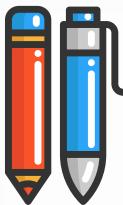


# Review once more



重温, 回顾 zhòng wēn , huí gù

- image - create and display image object
- imagesc - scale and display as image
- imshow - display image



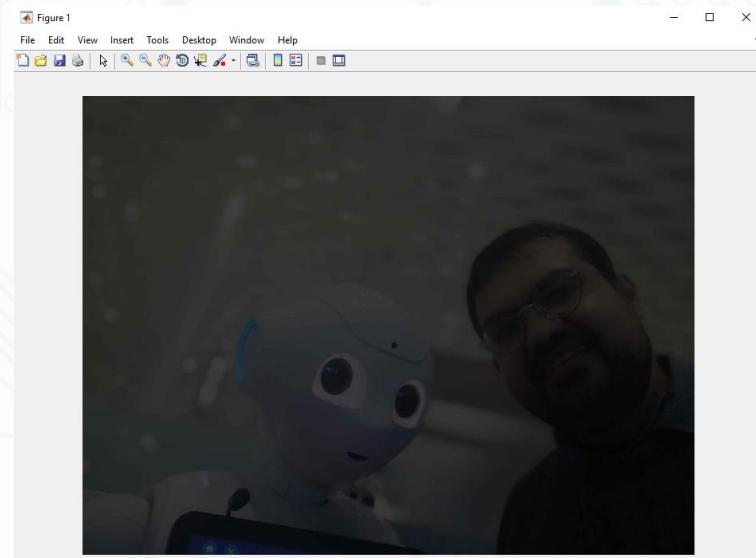
# A Single Grey Level

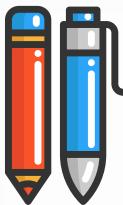
单一灰度



- Image quantisation can be described as a mapping process by which groups of datapoints (several pixels within a range of grey values) are mapped to a single point (a single grey level).
- An image can be re-quantised in MATLAB using the `grayslice` function.
- The following set of commands reduces the quantisation levels to 64 and displays the image:

```
clc  
close all  
  
A = imread('f:\ATA.jpg');  
  
B=grayslice(A, 64);  
  
imshow(B, gray(64))
```





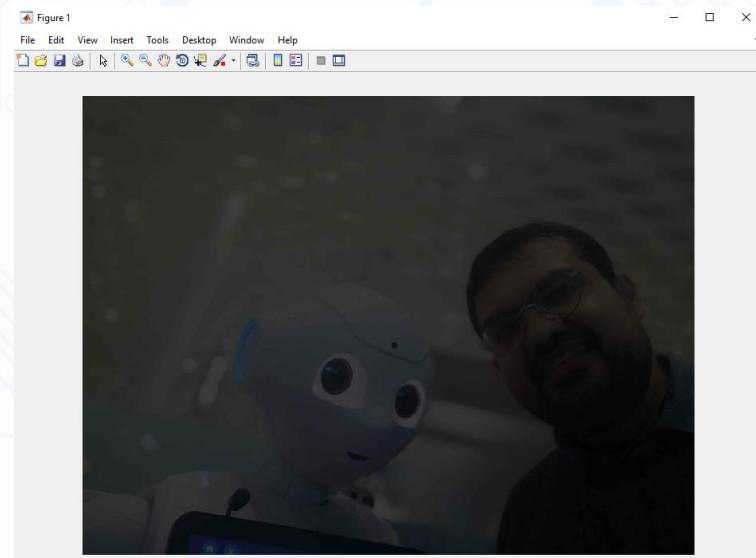
# A Single Grey Level

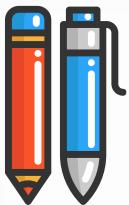
单一灰度



- 图像量化可以描述为一个映射过程，通过该过程，数据点组（灰度值范围内的几个像素）被映射到单个点（单个灰度级）。
- 可以在MATLAB中使用灰度切片函数对图像进行重新量化。
- 以下一组命令将量化级别降低到64，并显示图像：

```
clc  
close all  
  
A = imread('f:\ATA.jpg');  
  
B=grayslice(A, 64);  
  
imshow(B, gray(64))
```

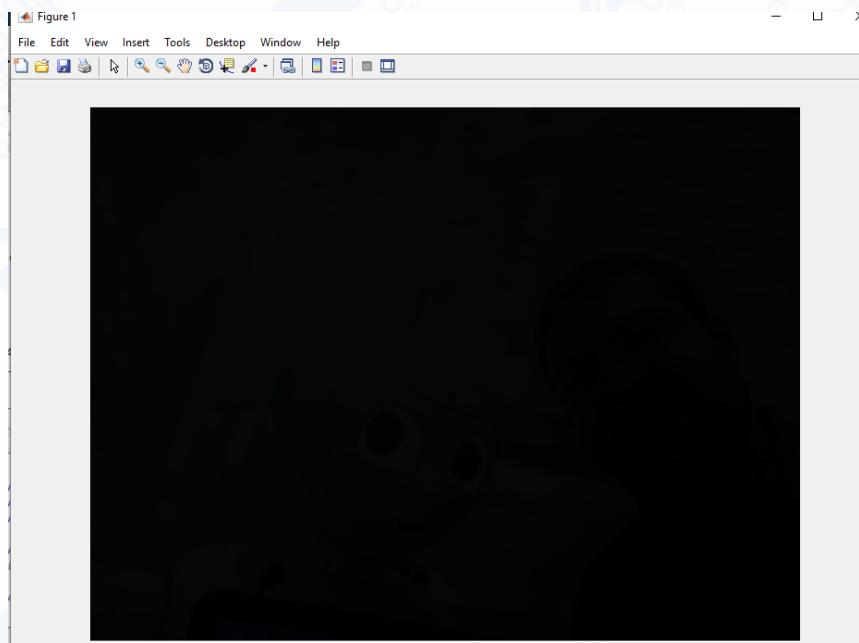




# A Single Grey Level



- The following set of commands reduces quantisation levels to 8 and displays the image:

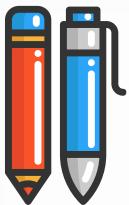


以下一组命令将量化级别降至8，并显示图像：  
以下一组命令将量化级别降至8，并显示图像：

```
clc  
close all
```

```
A = imread('f:\ATA.jpg');  
B=grayslice(A, 8);  
imshow(B, gray(8))
```



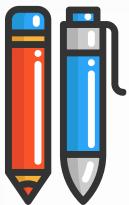


# Image Conversion Command

图像转换命令



gray2ind	intensity image to index image
im2bw	image to binary
im2double	image to double precision
im2uint8	image to 8-bit unsigned integers
im2uinti6	image to 16-bit unsigned integers
ind2gray	indexed image to intensity image
mat2gray	matrix to intensity image
rgb2gray	RGB image to grayscale
rgb2ind	RGB image to indexed image



# Histogram

直方图



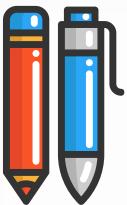
- Histogram of a greyscale image represents the frequency of its grey levels occurrence. It is a graph indicating the number of times each grey level occurs in the image.
- In a dark image, grey levels (and hence the histogram) are cluttered at the lower end. In a uniformly bright image, grey levels (and hence the histogram) are cluttered at the upper end.
- In a well contrasted image, grey levels (and hence the histogram) would be well spread out over much of the range.

灰度图像的直方图表示其灰度出现的频率。它是一个图形，指示图像中每个灰度级出现的次数。

在深色图像中，灰度（因此直方图）在下端杂乱无章。在均匀明亮的图像中，灰度（因此直方图）在上端杂乱无章。

在对比度良好的图像中，灰度（以及直方图）将在很大范围内分布。

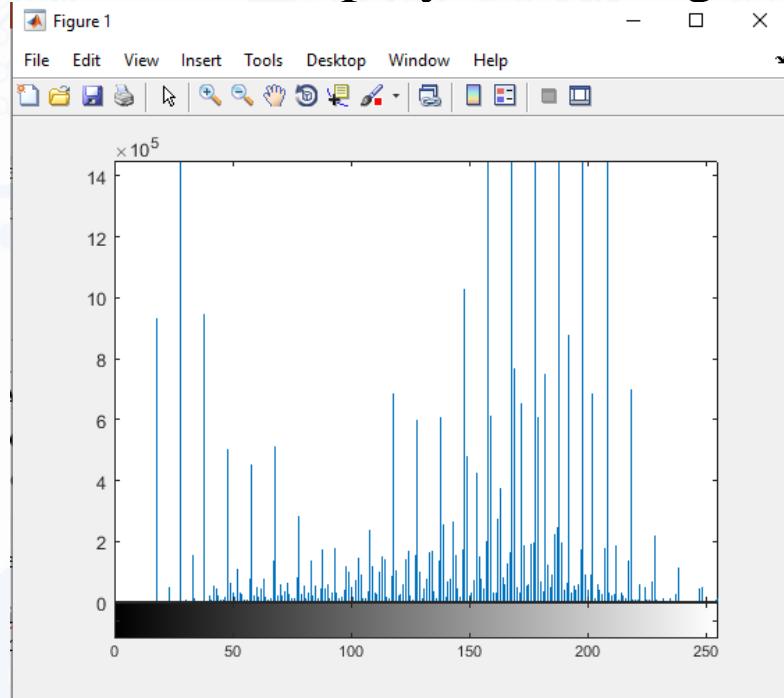
以下一组命令将量化级别降至8，并显示图像：



# Histogram



- In MATLAB, the histogram can be viewed using the `imhist` function. As an example, the commands that follow can be used to display the histogram of image

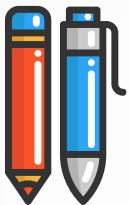


```
clc  
close all
```

```
A = imread('f:\ATA.jpg');  
figure(1), imhist(A);
```

The three main operations performed on a histogram include histogram stretching, histogram shrinking and histogram sliding. All these operations are described in the following paragraphs.

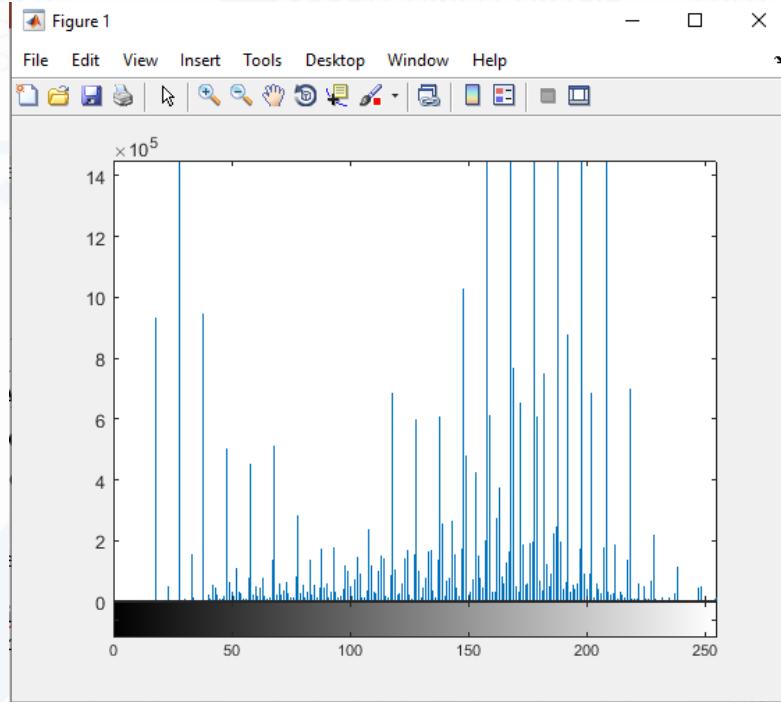




# Histogram



- 在MATLAB中，可以使用imhist函数查看直方图。例如，下面的命令可用于显示图像的直方图
- 以下一组命令将量化级别降至8，并显示图像：

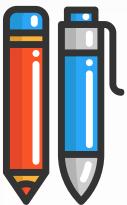


```
clc  
close all
```

```
A = imread('f:\ATA.jpg');  
figure(1), imhist(A);
```

对直方图执行的三个主要操作包括直方图拉伸、直方图收缩和直方图滑动。所有这些操作将在以下段落中描述。

以下一组命令将量化级别减少到8级并显示图像：对直方图执行的三个主要操作包括直方图拉伸、直方图收缩和直方图滑动。所有这些操作将在以下段落中描述。

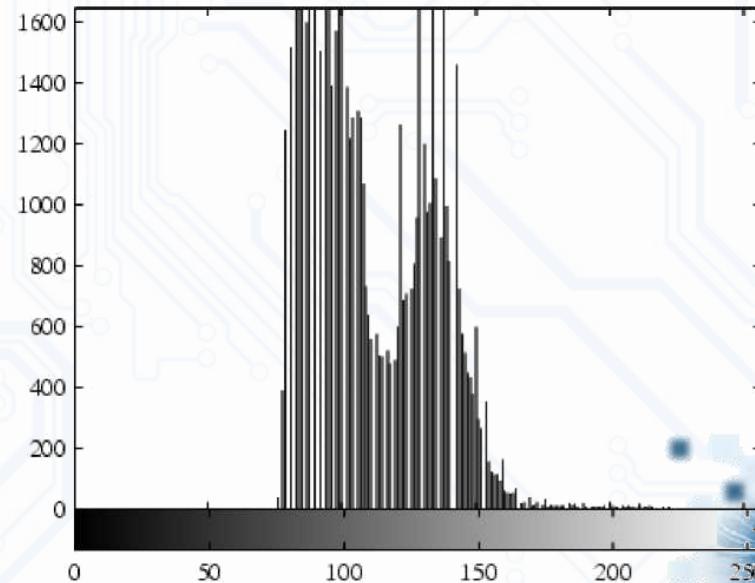


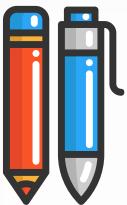
# imhist



As you can see, pout.tif is a somewhat low contrast image. To see the distribution of intensities in pout.tif in its current state, you can create a histogram by calling the imhist function. (Precede the call to imhist with the figure command so that the histogram does not overwrite the display of the image I in the current figure window.)

```
figure, imhist(I) % Display a histogram of I in a new figure.
```





# histeq

Notice how the intensity range is rather narrow. It does not cover the potential range of [0, 255], and is missing the high and low values that would result in good contrast.

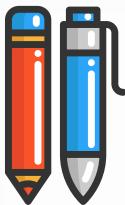
Now call **histeq** to spread the intensity values over the full range, thereby improving the contrast of I. Return the modified image in the variable I2.

```
I2 = histeq(I);  
% Equalize I and output in new array I2.
```

Display the new equalized image, I2, in a new figure window.

```
figure, imshow(I2)  
% Display the new equalized image I2.
```





# Histogram stretching

直方图拉伸

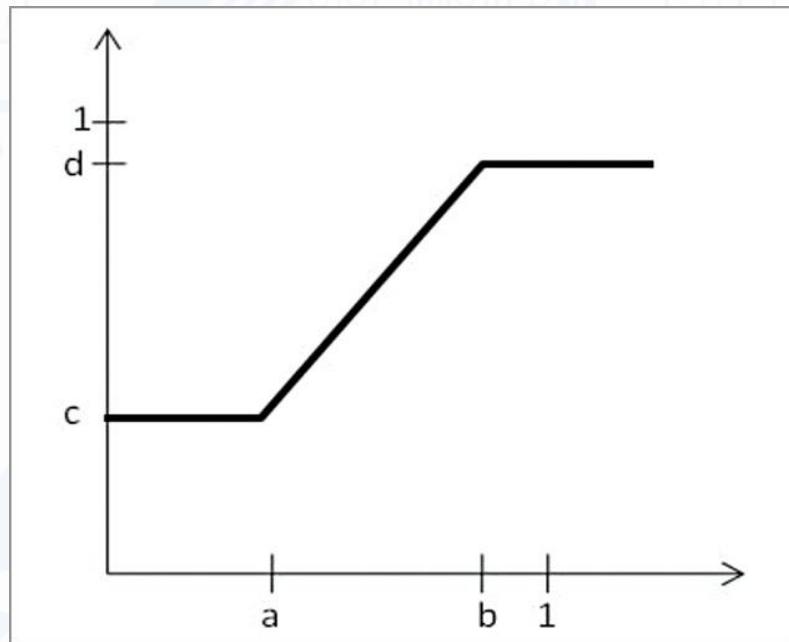


This technique, also known as input cropping, consists of a linear transformation that stretches part of the original histogram so that its non-zero intensity range occupies the full dynamic grey scale.

If the histogram of the image is cluttered at the centre, it can be stretched using `imadjust` function. The following command stretches the histogram as shown in Fig :

`imadjust (F, [a,b], [c,d])`

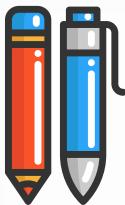
The values of a, b, c and d must be between 0 and 1.



Command `imadjust (F, [], [1,0])` inverts the grey value of the image, to produce a result similar to photographic negative.

Histogram stretching using above commands requires user input. An alternative approach is to use histogram equalisation command, which is entirely an automatic procedure.





# Histogram stretching

## 直方图拉伸

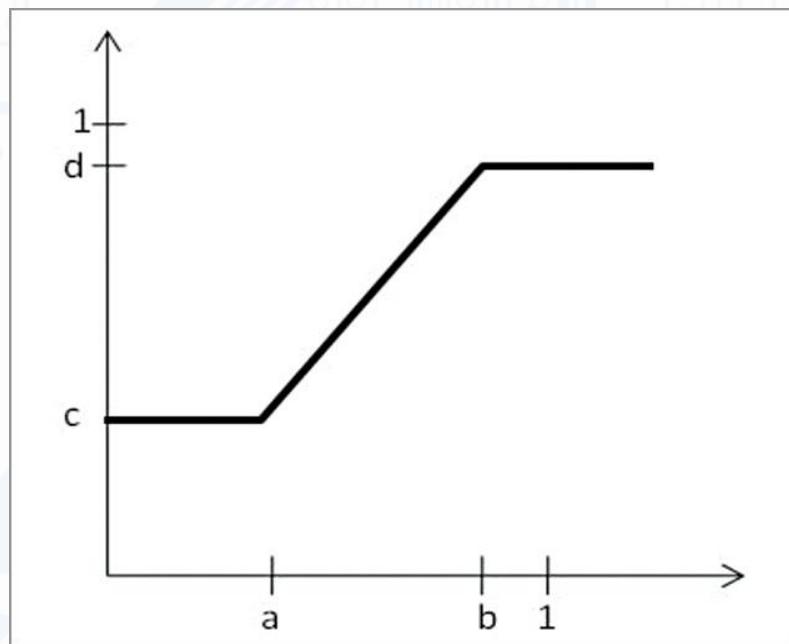


这种技术，也称为输入裁剪，由一个线性变换组成，该变换拉伸原始直方图的一部分，使其非零强度范围占据整个动态灰度。

如果图像的直方图在中心杂乱无章，则可以使用imadjust函数对其进行拉伸。以下命令拉伸直方图，如图所示：

`imadjust (F[a, b], [c, d])`

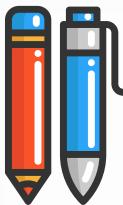
a, b, c和d的值必须介于0和1之间。



命令`imadjust (F, [], [1,0])`反转图像的灰度值，以产生类似于照相底片的结果。

使用上述命令进行直方图拉伸需要用户输入。另一种方法是使用直方图均衡命令，这完全是一个自动过程。





# Histogram stretching

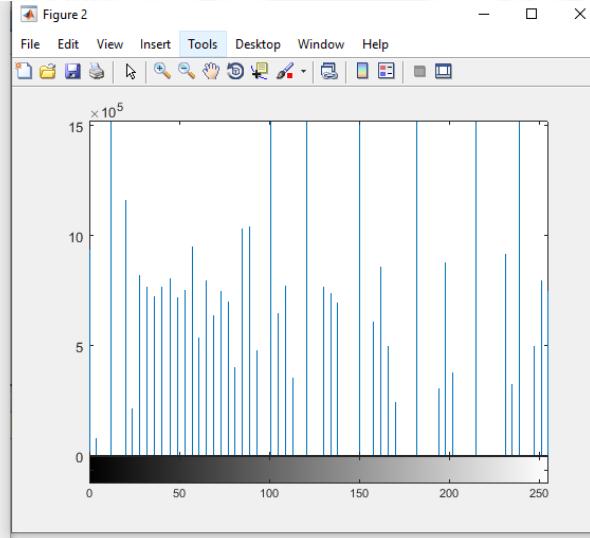
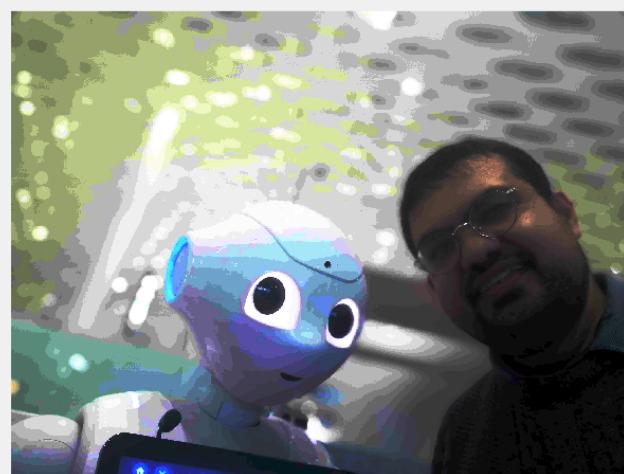


直方图拉伸

Histogram equalisation command in MATLAB is histeq.

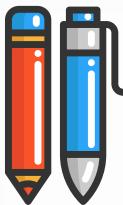
The use of histeq command is shown below:

```
clc  
close all  
  
A = imread('f:\ATA.jpg');  
HE = histeq(A);  
imshow(HE), figure, imhist(HE)
```



MATLAB中的直方图均衡命令是histeq。

histeq命令的使用如下所示：



# Histogram shrinking

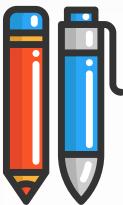
直方图缩小



- This technique, also known as output cropping, modifies the original histogram such that its dynamic greyscale range is compressed into a narrower greyscale. `imadjust` function can be used for histogram shrinking.

这种技术，也称为输出裁剪，修改原始直方图，使其动态灰度范围压缩为更窄的灰度范围。`imadjust`函数可用于直方图收缩。





# Histogram sliding



直方图滑动

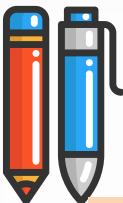
zhí fāng tú huá dòng

- This technique consists of simply adding or subtracting a constant brightness value to all pixels in the image. The overall effect is an image with comparable contrast properties, but higher or lower average brightness, respectively. `imadd` and `imsubtract` functions can be used for histogram sliding.
- When implementing histogram sliding, you must make sure that pixel values do not go outside the greyscale boundaries. An example of histogram sliding is given below:

这项技术只需向图像中的所有像素添加或减去一个恒定的亮度值。整体效果是具有类似对比度特性的图像，但平均亮度分别较高或较低。`imadd`和`imsubtract`函数可用于直方图滑动。

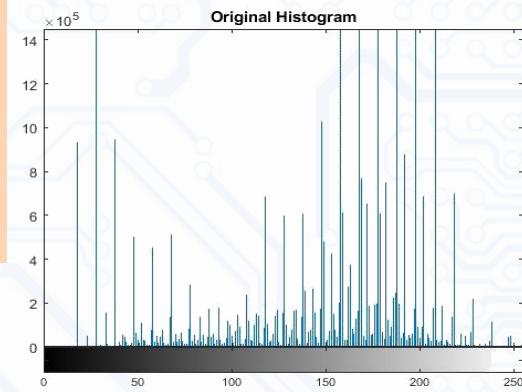
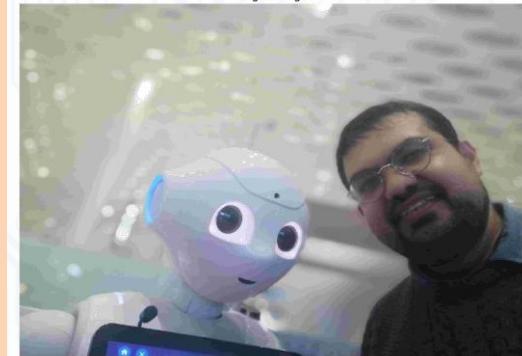
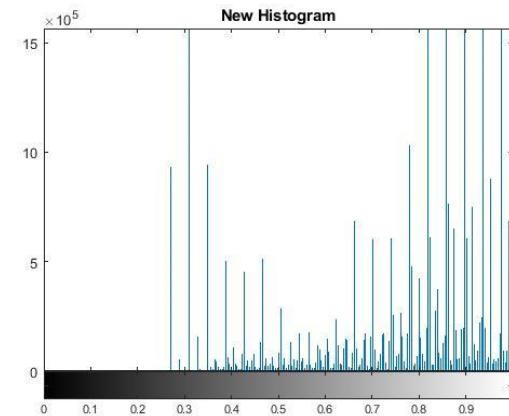
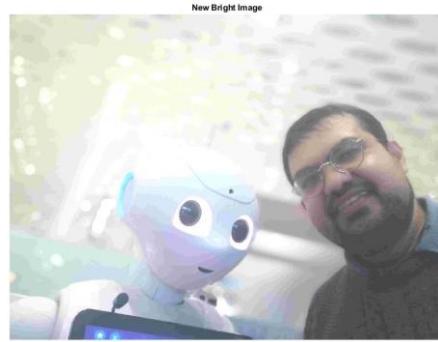
实现直方图滑动时，必须确保像素值不超出灰度边界。直方图滑动的示例如下所示：

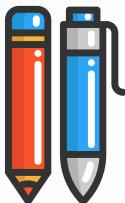




# Histogram .....

```
clc  
close all  
  
A = imread('f:\ATA.jpg');  
figure();imshow(A),title('Original Image');  
B=im2double(A);  
bright_add = 0.2;  
figure();imhist(A), title('Original Histogram');  
C=B+bright_add;  
figure();imshow(C),title('New Bright Image');  
figure();imhist(C), title('New Histogram');
```

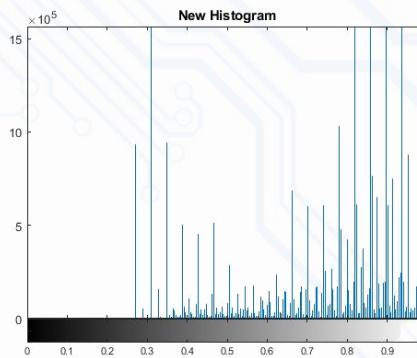
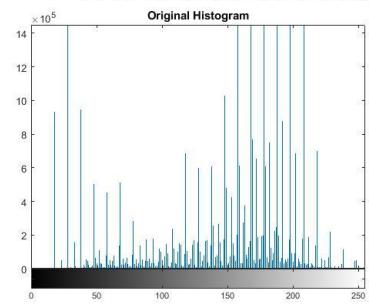
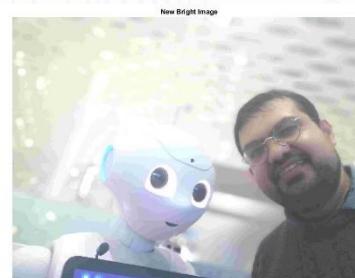
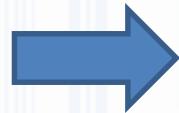
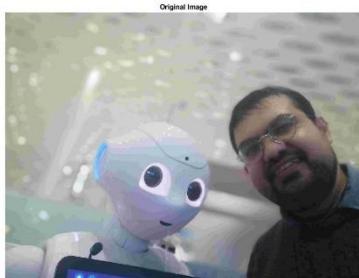




# Histogram .....



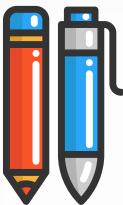
- In the example, the image is brightened by adding 0.2 to its pixel values.
- **Figs show the original image and its histogram, respectively.**
- **Figs show the modified image and its histogram, respectively.**



在本例中，通过向图像的像素值添加0.2，使图像变亮。

图分别显示了原始图像及其直方图。

图分别显示了修改后的图像及其直方图。



# Convert image to binary image

将图像转换为二值图像



## im2bw

Convert image to binary image, based on threshold

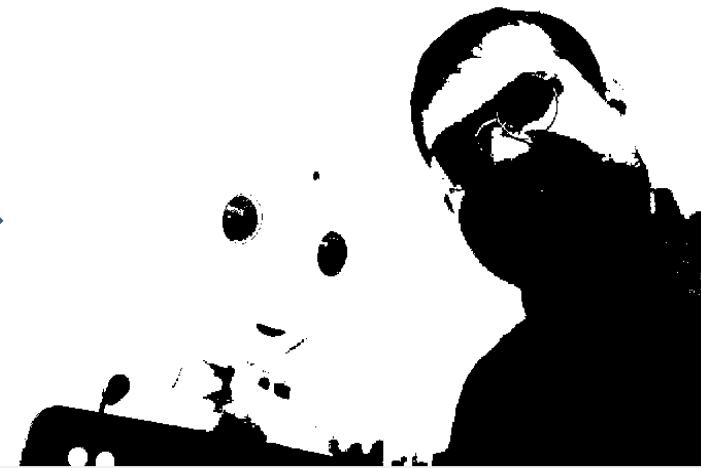
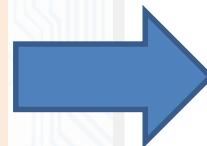
### Syntax

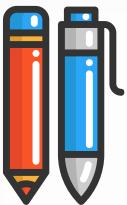
`BW = im2bw(I,level)`

`BW = im2bw(X,cmap,level)`

`BW = im2bw(RGB,level)`

```
clc  
close all  
  
A = imread('f:\ATA.jpg');  
BW= im2bw(A, 0.4);  
imshow(BW)
```





# Thresholding

阈值



- Thresholding is used to remove unnecessary details from an image and concentrate on essentials.
- It is also used to bring out hidden details, in case the object of interest and background have similar grey levels.
- Thresholding can be further classified as single thresholding and double thresholding.
- In MATLAB, single as well as double image thresholding can be done.

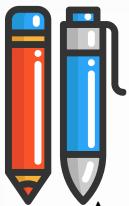
阈值化用于从图像中删除不必要的细节，并将注意力集中在要点上。

它还用于显示隐藏的细节，以防感兴趣的对象和背景具有相似的灰度。

阈值分割又可分为单阈值分割和双阈值分割。

在MATLAB中，可以进行单阈值和双阈值图像分割。





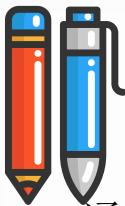
# Single thresholding

单阈值



- A greyscale image is turned into a binary image (black and white) by first choosing a grey level ‘T’ in the original image, and then turning every pixel black or white depending on whether its grey value is greater than or less than ‘T’.
- Thresholding is a vital part of image segmentation, where users wish to isolate objects from the background.
- To convert an image F into black-and-white image G with threshold of 100, the command in MATLAB is  $G=F>100$ .
- The following example reads image **Penguins\_grey.jpg** and displays both the original image and the image generated after thresholding using a factor of 70:  

```
>>A = imread('Penguins_grey.jpg');
>>imshow(A),figure, imshow(A>70)
```
- Figs show the original image and the image after thresholding, respectively. In addition, there is a command in MATLAB which converts a greyscale image or a coloured image into black-and-white image.
- The command is `im2bw(image,level)`, where `image` is the greyscale image and `level` is a value between 0 and 1.

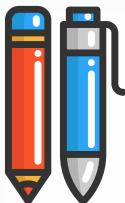


# Single thresholding

单阈值



- 通过首先在原始图像中选择灰度“T”，然后根据其灰度值是否大于或小于“T”，将灰度图像转换为二值图像（黑白）。
- 阈值分割是图像分割的一个重要部分，用户希望从背景中分离出对象。
- 要将图像F转换为阈值为100的黑白图像G，MATLAB中的命令为 $G=F>100$ 。
- 下面的示例读取图像Penguins\_grey.jpg，并显示原始图像和使用因子70进行阈值化后生成的图像：  
`>>a=imread ('Penguins_grey.jpg') ; >>imshow (A) 、figure、 imshow (A>70)`
- 图分别显示了原始图像和阈值化后的图像。此外，MATLAB中有一个命令，可将灰度图像或彩色图像转换为黑白图像。
- 该命令是`im2bw`（图像，级别），其中图像是灰度图像，级别是介于0和1之间的值。



# Finding the threshold value

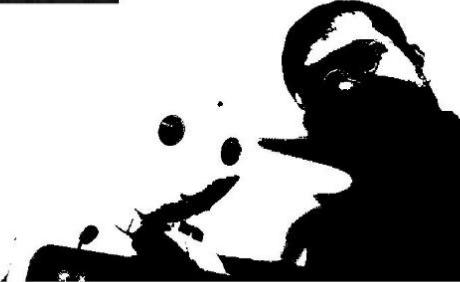


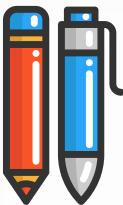
寻找阈值

xún zhǎo yù zhí



```
clc  
close all  
A = imread('f:\ATA.jpg');  
figure(); imshow(A);  
BW1=im2bw(A)  
figure(); imshow(BW1);  
BW2=im2bw(A, graythresh(A) )  
figure(); imshow(BW2)
```





# Use Morphological Opening to Estimate the Background



使用形态学打开来估计背景

- Notice that the background illumination is brighter in the center of the image than at the bottom. Use the `imopen` function to estimate the background illumination.

```
background = imopen(I,strel('disk',15));
```

To see the estimated background image, type

```
imshow(background)
```

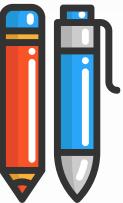
请注意，图像中心的背景照明比底部更亮。使用`imopen`功能估计背景照明。

```
background = imopen(I,strel('disk',15));
```

要查看估计的背景图像，请键入

```
imshow(background)
```

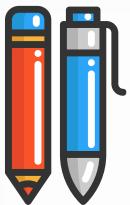




# Example

```
clc  
close all  
A = imread('f:\ATA.jpg');  
background1 = imopen(I,strel('disk',15));  
figure(1);imshow(background1)  
background2 = imopen(I,strel('disk',35));  
figure(2);imshow(background2)  
background3 = imopen(I,strel('disk',50));  
figure(3);imshow(background3)
```

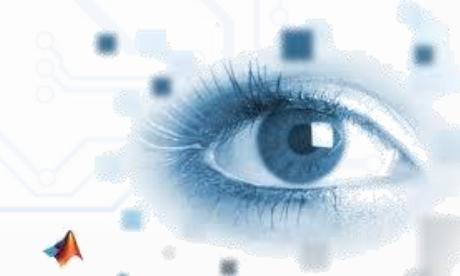


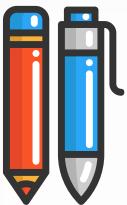


# You Try .....



```
clc  
close all  
a = imread('f:\ATA.jpg');  
imshow(a)  
b=imsharpen(a);  
figure(2), imshow(b)
```





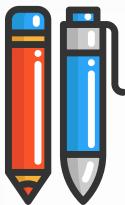
# Image sharpening

图像锐化 tú xiàng ruì huà



- Image sharpening is a powerful tool for emphasising texture and drawing viewer focus. It can improve image quality, even more than what is achieved through upgrading to a high-end camera lens.
- Most image sharpening software tools work by applying something called an ‘unsharp mask,’ which actually acts to sharpen an image. The tool works by exaggerating the brightness difference along the edges within an image. Note that the sharpening process is not able to reconstruct the ideal image, but it creates the appearance of a more pronounced edge.
- The command used for sharpening an image in MATLAB is:  
 $B = \text{imsharpen}(A)$





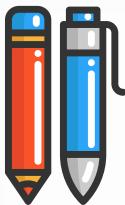
# Image sharpening

图像锐化 **tú xiàng ruì huà**



- 图像锐化是强调纹理和绘图查看器焦点的强大工具。它可以改善图像质量，甚至比升级到高端相机镜头所取得的效果还要好。
- 大多数图像锐化软件工具都是通过应用一种叫做“反锐化遮罩”的东西来工作的，这种遮罩实际上起到了锐化图像的作用。该工具通过放大图像中沿边缘的亮度差来工作。请注意，锐化过程无法重建理想图像，但它会创建更明显边缘的外观。
- 在MATLAB中用于锐化图像的命令是：  
 $B=\text{imgipe} (A)$

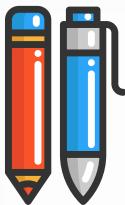




# Image sharpening



- The command used for sharpening an image in MATLAB is:  
 $B = \text{imsharpen}(A)$
- It returns an enhanced version of the greyscale or the true-colour (RGB) input image A, where image features such as edges have been sharpened using the unsharp masking method.
- $B = \text{imsharpen}(A, \text{Name}, \text{Value}, \dots)$  sharpens the image using name-value pairs to control aspects of unsharp masking.
- Let us see the use of imsharpen function:

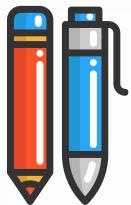


# Image sharpening



- 在MATLAB中用于锐化图像的命令是：  
 $B=\text{imgrip}(A)$
- 它返回灰度或真彩色（RGB）输入图像A的增强版本，其中图像特征（如边缘）已使用反锐化掩蔽方法锐化。
- $B=\text{imgrip}(A, \text{名称}, \text{值}, \dots)$  使用名称-值对来控制取消锐化掩蔽的各个方面来锐化图像。
- 让我们看看函数的用法：

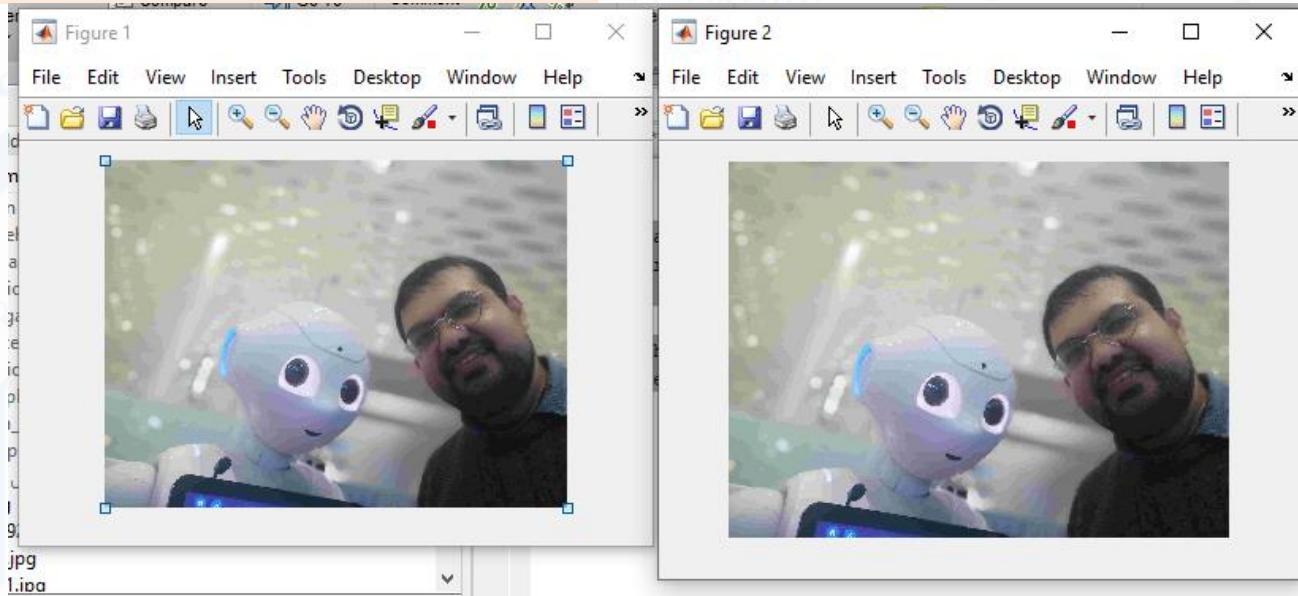


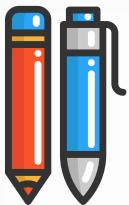


# Image sharpening



```
clc  
close all  
a = imread('f:\ATA.jpg');  
Figure(1);imshow(a)  
b=imsharpen(a);  
figure(2),imshow(b)
```





# Image sharpening

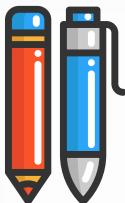
图像锐化



- You can specify ‘radius’ and ‘amount’ parameters in the imsharpen function as given in the example below:
- $b=imsharpen(a, 'Radius', 4, 'Amount', 2);$   
*figure, imshow(b)*

您可以在imgripe函数中指定“半径”和“数量”参数，如下例所示：





# Display the Background Approximation as a Surface

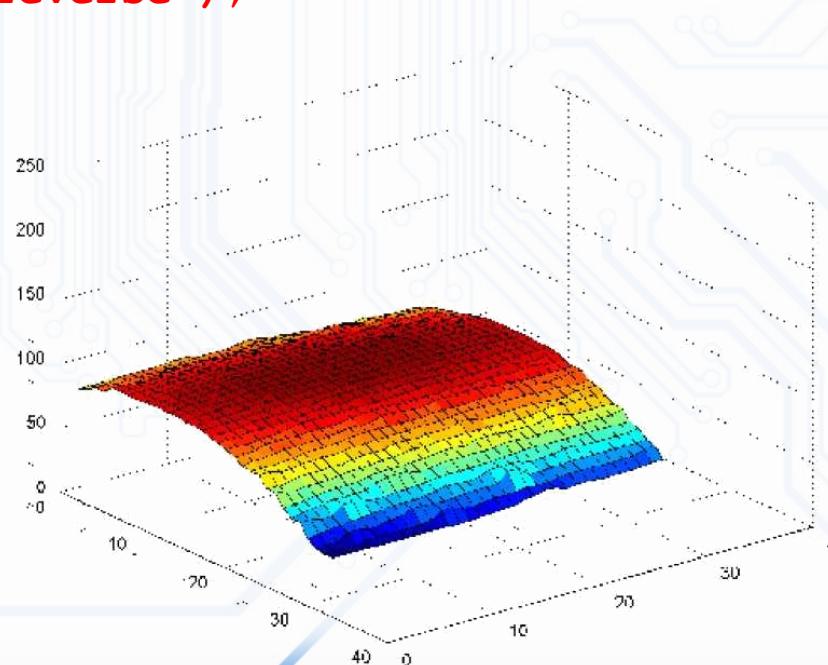


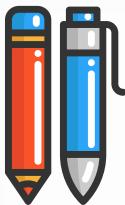
将背景近似显示为曲面

Use the `surf` command to create a surface display of the background approximation, background. The `surf` function requires data of class `double`, however, so you first need to convert background using the `double` command.

使用`surf`命令创建背景近似的曲面显示，即背景。但是，`surf`函数需要类`double`的数据，因此首先需要使用`double`命令转换背景。

```
figure, surf(double(background(1:8:end,1:8:end))), zlim([0 255]);  
set(gca,'ydir','reverse');
```

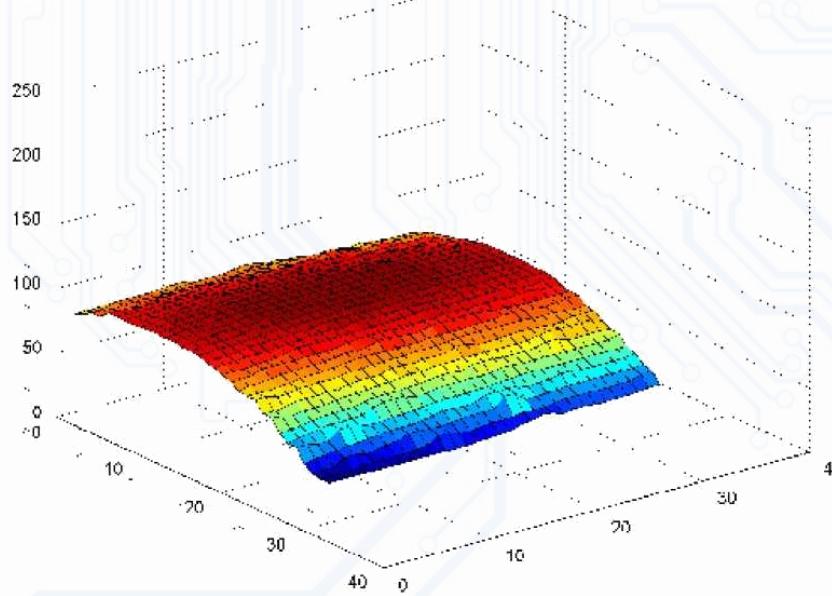




# Display the Background Approximation as a Surface

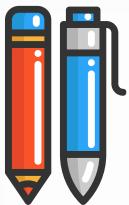


- The example uses MATLAB indexing syntax to view only 1 out of 8 pixels in each direction; otherwise the surface plot would be too dense. 该示例使用MATLAB索引语法来查看每个方向上8个像素中的1个；否则，曲面图将过于密集。
- The example also sets the scale of the plot to better match the range of the `uint8` data and reverses the y-axis of the display to provide a better view of the data (the pixels at the bottom of the image appear at the front of the surface plot).

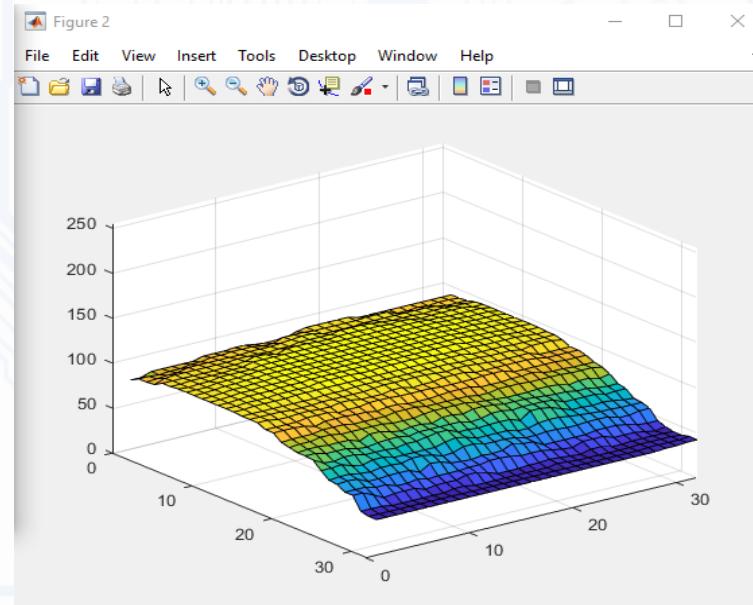
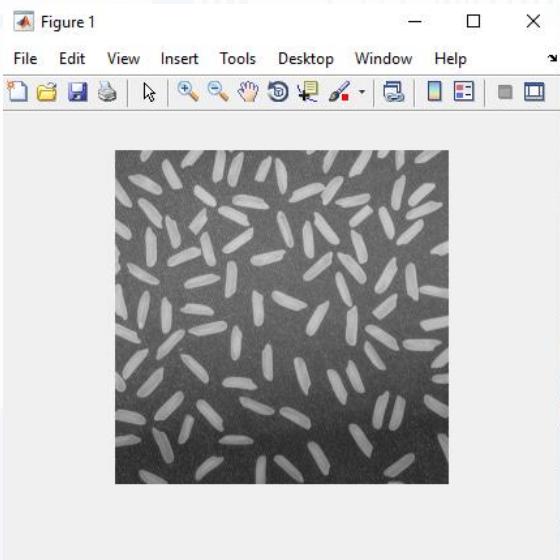


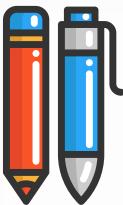
该示例还设置了绘图比例，以更好地匹配`uint8`数据的范围，并反转显示器的y轴，以提供更好的数据视图（图像底部的像素显示在曲面绘图的前面）。





```
I = imread('rice.png');
imshow(I)
background = imopen(I,strel('disk',15));
figure
surf(double(background(1:8:end,1:8:end))),zlim([0 255]);
set(gca,'ydir','reverse');
```





# Student Task\_3: DIP



- 请帮我翻译部分的朋友鼓掌
- Qǐng bāng wǒ fānyì bùfèn de péngyǒu gǔzhǎng

## Solve the Question shared in mooc

解决mooc分享的问题

Repeat all the example in this PPT with your personal photo

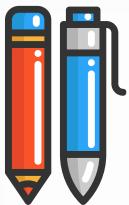
And list the function which use with description

Please send your task based on template



**Send for Next lecture**

**发送下一个讲座**



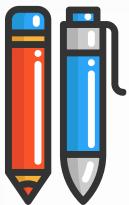
# Reference



Introduction to MATLAB, *Kadin Tseng, Boston University,  
Scientific Computing and Visualization*

- Images taken from Gonzalez & Woods, Digital Image Processing (2002)





# Reference



1. [www.mathworks.com/products/\*\*image\*\*/](http://www.mathworks.com/products/image/) [www.mathtools.net/MATLAB/\*\*Image Processing\*\*/](http://www.mathtools.net/MATLAB/ImageProcessing/)
2. [www.amath.colorado.edu/courses/4720/2000Spr/Labs/Worksheets/\*\*Matlab tutorial\*\*/\*\*matlabimpr\*\*.html](http://www.amath.colorado.edu/courses/4720/2000Spr/Labs/Worksheets/Matlab tutorial/matlabimpr.html) [www.imageprocessingplace.com/DIPUM/dipum book description/book description.htm](http://www.imageprocessingplace.com/DIPUM/dipum book description/book description.htm)



江西理工大学

Jiangxi University of Science and Technology

信息工程学院

School of information engineering

## Digital Image Processing

THANK YOU





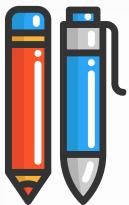
“I didn’t do my work for money or prizes - only for the excitement of discovery.”

EDMUND PHELPS  
Prize in Economic Sciences 2006

MAY-BRITT MOSER  
Nobel Prize in Physiology or Medicine 2014

“I learned at an early age that  
work makes you happy.”





**“BE HUMBLE. BE HUNGRY.  
AND ALWAYS BE THE  
HARDEST WORKER  
IN THE ROOM.”**

