

Digitizing Details

Creating your own digital images



Lesson

We use scanners, digital cameras, and printers to produce digital images for everything from advertising to archiving documents to conducting scientific research. By understanding the basic aspects of the digitizing process, you are more likely to produce the images you want without losing important information.


A digital image is stored as a series of numerical values. When these values are arranged in a rectangular grid of rows and columns, and a specific color is assigned to each different value, a picture is formed. Each point in the grid is called a pixel, or *picture element*. The numerical values of the pixels may represent the various colors of the image, as in a paint-by-number picture, or they may represent measurements such as temperature, elevation, or reflectance.

Three fundamental and interrelated characteristics of digital images are *bit depth*, *spatial resolution*, and *file format*. When working with digital images, the key question is “What do you intend to do with this image?” Will you display it on a computer screen, or do you need to print the image? Keeping your purpose in mind will help you to make decisions about image characteristics and achieve the result you want.



Bit Depth (depth resolution)

Bit depth refers to the number of bits, or binary digits, used to store the value of each pixel in an image. The number of bits determines the range of possible pixel values. Each unique pixel value represents a specific color, so bit depth also represents the number of possible colors in the image.

 Open all the images in the **Bit Depth** folder, starting with **Sue (1-bit).tif** and ending with **Sue (8-bit).tif**. (Do not open **Sue (1-bit) dithered.tif** yet.)

 Choose **Window > Tile** to arrange the eight images so you can see them all at the same time.

1. Describe how the appearance of the image changes as the bit depth decreases from 8 bits to 1 bit.

 Click , then click twice on the image to zoom in 400% on the face in each image. Hold down the **Shift** key as you click on each image to keep the window from changing size as you zoom.

2. At what bit depth does the image quality begin to degrade noticeably?

A *bit*, or binary digit, can have a value of either 0 or 1. Thus, each pixel in a 1-bit digital image can be either black (0) or white (1). In a 2-bit image, each pixel is represented by 2 bits, each of which can have a value of 0 or 1. This produces four possible pixel values: 00, 01, 10, and 11, which correspond to four colors, such as black, dark gray, light gray, and white.

Archiving documents

In the past, banks, utility companies, and other businesses processed and stored paper copies of documents, such as contracts, invoices, and cancelled checks. Today, these documents are often scanned and stored electronically, saving significant time, money, and storage space.

Grays are colors too!

In this text, we will refer to any visual representation of a pixel value as a *color*, so all shades of gray are colors, as are black and white!

“Talking” binary

The numbers we use every day are decimal, or “base ten” numbers. When we write binary, or “base two” numbers, we add a subscript of 2 (the number base), so the binary value 10 would be 10_2 .

When you read or say a binary number, say each digit separately, as either “zero” or “one”. For example, read 10_2 as “one-zero,” not “ten” and 11_2 as “one-one,” not eleven. In the text, binary numbers are written without the base subscript.

Rather than figure out all the possible combinations of binary digits for images of different bit depths, you can easily calculate the number of possible pixel values by raising 2 to the power of the number of bits per pixel.


- Complete the table showing the number of possible colors that correspond to the following bit depths.

Bit depth	Expanded Representation	Exponential Representation	Possible Colors
1	2	2^1	2
2	2×2	2^2	4
3	$2 \times 2 \times 2$	2^3	
4	$2 \times 2 \times 2 \times 2$	2^4	
5	$2 \times 2 \times 2 \times 2 \times 2$	2^5	
6	$2 \times 2 \times 2 \times 2 \times 2 \times 2$	2^6	
7	$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$	2^7	
8	$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$	2^8	

- Based on the table, what is the maximum number of colors that the **Sue (4-bit).tif** image can contain?


 Activate the **Sue (4-bit).tif** image, and choose **Analyze > Histogram**.

- According to its histogram, how many different pixel values, or colors, does the 4-bit image contain? (Count the number of lines, or “spikes,” in the histogram. Don’t forget to count the spikes at each end of the histogram.)

 Close the 4-bit histogram window. Activate and create a histogram of each of the other bit depth images. Examine each histogram briefly before closing it.

 Close all the histogram and image windows.

 Re-open the **Sue (1-bit).tif** image, and open the **Sue (1-bit) dithered.tif** image.

 Activate and create a histogram of each image.

According to the histograms, both images contain only black and white pixels, yet they look very different—the dithered image looks somewhat more realistic. *Dithering* is a process that arranges pixels of the available colors in patterns to create the illusion of smoother or additional colors. For example, alternating black and white pixels appear as a medium gray.

 Close all the histogram and image windows.

Image types

ImageJ can open, save, and convert between five different types of images. In this section, you will examine and compare these image types. These types are defined by bit depth and whether they are color or grayscale images. In addition to comparing the images’ appearance and pixel values, you will examine the amount of memory ImageJ uses to hold each type of image.

TIFF images in ImageJ

Technically, TIFF (.tif) images can have almost any bit depth. Like most software, however, ImageJ only supports certain standard depths: 1-bit, 8-bit, 16-bit, 32-bit, 8-bit RGB, and 16-bit RGB images.

In the **Sue** images, the bit depths are simulated—each is actually an 8-bit TIFF image containing a limited number of colors.

ImageJ and dithering

ImageJ does not produce dithered images because it is a scientific tool, and dithering is an artistic technique for producing “prettier” images from a limited number of colors.

The image used here was produced by GraphicConverter, a shareware image processing application for Macintosh computers.





Like most other graphic applications, ImageJ stores and processes image data in groups of 8 bits, called *bytes*, abbreviated *B*. Depending on the type of image, pixel values are stored as 1, 2, 3, 4, or 6 bytes. Obviously, the number of bytes per pixel influences both the amount of working memory and the amount of storage required for the image data. The table below shows the number of possible pixel values for each image type.

Bytes per pixel	Bits per pixel	Number of possible pixel values	Range of pixel values
1	1 × 8 = 8	2 ⁸ = 256	0–255
2	2 × 8 = 16	2 ¹⁶ = 65,536	0–65535
3	3 × 8 = 24	2 ²⁴ = 16.7 million	Stay tuned to see how these work!
4	4 × 8 = 32	2 ³² = 4.3 billion*	
6	6 × 8 = 48	2 ⁴⁸ = 281 trillion!	

Memory is usually measured in *kilobytes* or *megabytes* (or sometimes in *gigabytes*). The prefix kilo- usually means “one thousand,” but since we’re working in binary numbers, kilo- represents 2¹⁰ or 1024. Mega- (millions) means 2²⁰ and giga- (billions) means 2³⁰.

So, to convert from bytes to kilobytes, from kilobytes to megabytes, or from megabytes to gigabytes, divide by 1024.

8-bit images

-  In the **Bit Depth (types)** folder, open the **Balloon 8-bit.tif** image.
-  Move your cursor over the image, and watch the changing pixel value in the ImageJ window status bar. Notice that the pixel values are all within the range of 0–255.

8-bit images are grayscale images that use one byte per pixel. To calculate the size of the image, multiply the number of pixels by 1 byte per pixel. The number of pixels in an image is found by multiplying the width by the height of the image, both in pixels (abbreviated *px*).

$$\text{Width} \times \text{Height} \times \text{Bit Depth} / 1024 = \text{Memory Required (KB)}$$

$$420 \text{ px} \times 600 \text{ px} \times 1 \text{ B/px} \div 1024 = 246 \text{ KB}$$

The amount of memory required for each image is the last value displayed in the status bar of each image window. The memory size is also listed after the image name at the bottom of the **Window** menu.

6. According to the image window status bar, what is the size of the **Balloon 8-bit.tif** image, in kilobytes (K)? How well does this match your calculated value?

-  Close the **Balloon 8-bit.tif** image.

16-bit images

-  Open the **Balloon 16-bit.tif** image.

This grayscale image of the balloon stores 16 bits or 2 bytes for each pixel. Move your cursor over the image and watch the ImageJ status bar to see that the pixel values are all between 0 and 65535.

7. This image has the same width and height as the 8-bit version. Without peeking, predict how much memory you think this image will occupy.

* 32-bit images

In ImageJ, 32-bit grayscale images are also called *floating point* images. Floating point values are decimal approximations of real numbers and range from 3.402×10^{38} to -1.175×10^{37} , a truly astonishing range of possible values, and well beyond the 4.3 billion stated in the table!

“32-bit float” images are particularly valuable in scientific applications because they can store a wide range of measurements with little or no loss of information.

8. Compare your prediction with the actual memory size reported in the image window.

 Close the **Balloon 16-bit.tif** image.

32-bit images

 Open the **Balloon 32-bit.tif** image.

This image type uses 4 bytes per pixel and has the same width and height as the previous two images.

9. Predict the size, in kilobytes, of this image, then compare it to the actual memory size reported in the image window.

 Close the **Balloon 32-bit.tif** image.

 Open the **Cygnus A 32-bit.tif** image.

This image was produced from data collected by a radiotelescope—a large antenna designed to receive radio signals from objects in space. The object shown in this image is a powerful radio galaxy named Cygnus A, located in the constellation Cygnus (the swan). The two light blobs at the edges of the image are jets of material being ejected from the galaxy.

 Move your cursor over the **Cygnus A 32-bit.tif** image, and note the pixel values.

In 32-bit grayscale images, the value of each pixel is represented by a *32-bit floating point* number (see sidebar at right). The pixel values are real numbers representing the strength of the radio signal in units called *Janskys*, after Karl Jansky, the father of radioastronomy.

 Choose **Image > Adjust > Window/Level**, and click the **Auto** button in the **W&L** panel. Close the **W&L** panel.

The image's lookup table is now spread over the range of values in the image, revealing detail in the two gas jets as well as their source, the compact bright object in the center. You can also apply a false-color lookup table to make these features easier to see (or just more attractive to look at).

 Choose **Image > Lookup Tables > Fire**.

10. What is an advantage of 32-bit floating point image data?


 Close the **Cygnus A 32-bit.tif** image.

8-bit RGB color images

So far, you have been looking at grayscale images. These images have only one color *channel*, which represents brightness or some other measurement. In ImageJ, color images have three separate color channels, one for each of the three primary colors of light—Red, Green, and Blue (RGB). The color of each pixel in the image is determined by the *combination* of these three color channels.

 Open the **Balloon RGB color.tif** image.

In 8-bit RGB color images, each channel has 8 bits (one byte) per pixel in each channel. Since there are three channels, it takes a total of three bytes (24 bits) to describe each pixel.

 Move your cursor around the image, and note the changing pixel values in the ImageJ window.

Why do the 8-, 16-, and 32-bit images all look the same?

Since your eye can only differentiate between about 200 different shades of gray, it would be pointless to try to display more than 256 different grays. All three types of images use a 256-color lookup table. With 8-bit images, each different value has a unique color. With 16- and 32-bit images, each different color represents a *range* of pixel values.

The advantages of greater bit depth come not from the appearance of the image, but from the increased scientific and analytical value of the data.

32-bit floating point

Most image types used by **ImageJ** contain integer pixel values—positive and negative whole numbers. In 32-bit grayscale images, pixel values are *floating point* numbers—decimal values that represent real numbers.

The advantage of floating point notation is that pixel values can range from very large numbers to very small numbers (both positive and negative), including fractions.

The 32 bits are assigned as follows:

bit(s)	meaning
1	sign
2-8	exponent
9-32	significand

You may be familiar with scientific notation, which represents the value 74,089.5 as 7.40895×10^4 . 32-bit floating point notation is simply *binary* scientific notation. The *significand*, like the mantissa in scientific notation, contains the 6 or 7 most significant digits of the number, in binary format.

The range of possible values for 32-bit floating point images is very impressive:

Largest	$\sim 3.4 \times 10^{38}$
Smallest	$\sim 1.18 \times 10^{-38}$

The ImageJ status bar lists three separate values for each pixel—a red, green, and a blue channel value, in that order.



11. Assuming that this image has the same width and height as the other balloon images, predict its size, in kilobytes. Compare your prediction with the actual memory size reported in the image window.

This time, your prediction was probably wrong. This is because ImageJ works with RGB pixels as 32-bit integer data. Each pixel requires one byte for each color channel, plus one unused byte—a total of four 8-bit bytes per pixel. The fourth byte is called the *alpha channel*, which is not used by ImageJ. Other image processing software uses the alpha channel for masking or transparency information when blending two or more images. ImageJ discards the alpha channel when it saves the image data, so the saved RGB file is smaller than the same file when it is open.

16-bit RGB color images


-  Open the **Balloon RGB stack.tif** image.

Some digital cameras and scanners can produce RGB color images with 16 bits (2 bytes) per channel, for a total of 48 bits of data per pixel. Most computers and monitors are unable to display images with this bit depth, so ImageJ opens them as stacks consisting of three 16-bit grayscale color channels. In ImageJ, you can process one or all three slices of the stack to analyze or modify the image data.

-  Examine the three color channels by dragging the slider at the bottom of the stack window, and observe the slice counter in the window status bar.
-  Move your cursor over the image, and watch the ImageJ window status bar to convince yourself that the pixel values in each channel are 16-bit values. Although you don't see colors, all the data are there, available for analysis.

The width and height of this image are the same as for the other balloon images, but now each pixel requires three channels with two bytes per channel—a total of six bytes or 48 bits per pixel.


12. Calculate the total amount of memory required for this image, in *megabytes*. (To convert from kilobytes to megabytes, you must divide the size in kilobytes by 1024!) Compare your prediction to the actual memory size reported in the image window status bar.

-  Convert the 16-bit RGB image to an 8-bit RGB color image by choosing **Image > Color > Convert Stack to RGB**.

Note that the memory used by this 8-bit RGB color image is the same as the one you opened earlier.

8-bit color images

It is possible to convert an 8-bit RGB image to an 8-bit color image without separate color channels. To display such an image, ImageJ must create a custom lookup table containing 256 colors used to “paint” the image. ImageJ then sets the value of each pixel so that it references or *indexes* the appropriate color from the lookup table for that location in the image. For this reason, 8-bit color images are also called *indexed color* images.

-  To convert the 8-bit RGB image to an 8-bit indexed color image, choose **Image > Type > 8-bit Color**.

Open versus saved image sizes

When calculating image sizes, the size of the open image (in kilobytes) can be calculated as:

$$\frac{\text{width (px)} \times \text{height (px)} \times \text{bit depth} \times \text{number of channels}}{1024 \text{ bytes/kilobyte}}$$

When you *save* an image, however, the image file will often be a different size than the open image. This is due to several factors:

- Some image formats use coding and compression techniques to reduce the size of the file.
- When a computer saves a file to certain storage media, data are written in chunks of a specific size, called *sectors*. The image data may not fit exactly into a whole number of sectors, so the last sector may contain unused space that adds to the file size.
- Some image file formats save additional data in the file along with the actual image data—*tags, headers, metadata*, etc.

TRILLIONS of colors!

Though not infinite, the number of available colors in 48-bit images is astonishing.

$$2^{16} \times 2^{16} \times 2^{16} = 281 \text{ trillion colors!}$$

Even commercial applications such as Adobe Photoshop do not display the full 48-bit color depth on screen. The images are displayed in 24-bit color, but all processing is done on the 48-bit pixel data.

13. What happens to the quality of the image when you convert it to indexed color? In which part(s) of the image are the colors most poorly reproduced?

As explained earlier, ImageJ does not use dithering to create smoother-looking colors at lower bit depths. Here is another example.

 Open the **Balloon 8-bit color.tif** image.

This 8-bit indexed color version of the image was produced using software that uses dithering to increase the apparent number of colors in the image.


14. In general, what happens to amount of memory required for an image as the bit depth increases?

 Close all the open image windows.

Spatial resolution

Spatial resolution refers to the number of measurements, or *samples*, per unit width or length of an image. Spatial resolution is usually expressed in dots per inch (dpi) or pixels per inch (ppi). For example, current flatbed scanners sample, or measure, the light reflected from the original material up to 2400 dpi or even higher. Let's compare images of a photo scanned at different resolutions.


 Open the five images in the **Spatial Resolution** folder.

 Without changing the sizes of the windows, arrange the images so you can see them all at the same time.

These images were scanned at 8-bit grayscale pixel depth and spatial resolutions ranging from 38 to 600 dpi. Each image has roughly twice the resolution of the previous image.



15. What happens to the size of an image as its resolution increases?
16. What are the dimensions of the 300-dpi image?
17. Calculate how much memory this image would require. Remember, memory size equals:

$$\frac{\text{Width (px)} \times \text{Height (px)} \times \text{Bit depth (bytes/px)} \times \text{Color Channels}}{1024 \text{ bytes/kilobyte}}$$




 The memory occupied by each image is displayed at the bottom of the Window menu, as well as on the status bar of each image window.

18. What happens to the memory size of the image as you double the spatial resolution?
19. Predict the size of this image if you increased the resolution to 1200 dpi.

When saving an image, the spatial resolution of the image affects its file size, so you must think about how you will be storing and transporting your image. If you want to e-mail an image to a friend who has a slow modem connection, then you want as small a file size as possible. If you are sending the image on a CD-ROM, then the image can be quite large.

 Use  to magnify or reduce the images until they are the same size as the 300 dpi image. (Hold down the **Alt** or **Option** key to reduce the 600-dpi image to 50% scale.)



-  Choose **Window > Tile** to tile the images so you can see them all at the same time.
-  Hold down the **Shift** key, and use  to click on each image until the girl's face fills the window. (For the 300 dpi image, this would be at a scale of 400%, as displayed on the image window title bar. Holding down the **Shift** key prevents the window from changing size while you zoom.)

20. How do the details in the image change as the resolution increases?

An important constraint to spatial resolution is file size. Higher resolution provides more detail, but with it comes increasing file size. In fact, you have seen that the file size increases as the *square* of the increase in resolution. Thus, if you double the resolution, the file size increases to *four* times its original size.

21. If you scan an image at 100 dpi and the file size is 50K, what will the file size be if you increase the scan resolution to 300 dpi?

If you are creating a series of images to use in a stack, you may want to choose lower resolutions to keep the overall stack size small.

Resolution for display and printing

In addition to considering the input resolution (from a scanner or camera), it is also important to consider the resolution of the output device. For instance, if you want an image displayed on-screen to be roughly the same size as the original photograph, then you will need to scan the image at the same spatial resolution as the monitor. If a monitor displays 100 pixels per inch, an image scanned at 100 dpi will appear approximately life-size when displayed on that monitor at 100% magnification.

Printing images is considerably more confusing because printers usually state their resolution in dots per inch, but printing quality is actually related to a different measure related to the printing device, *lines per inch (lpi)*.

For example, a “high resolution” 1440-dpi inkjet printer prints with a resolution of only about 240–300 lpi. A good rule of thumb for determining the ideal image resolution is:

$$\text{dpi} = 1.5 \times \text{lpi}$$

Thus, a scanned image with a resolution of 360 to 450 dpi would be more than sufficient to print at highest quality on this printer. Any additional resolution will be ignored by the printer, and will slow down the printing process.

22. Using the output device resolution table at right, what would be the ideal image resolution for printing on a 600-dpi laser printer?

-  Close all the open images.

LPI values for common output devices

Printer or Output	LPI
Laser printer, 300 dpi	53–60
Laser printer, 600 dpi	60–106
Newspaper	65–100
Book (uncoated paper)	120–133
Book (coated paper)	133–150
Color Magazine	150–175
Art Book and Magazine	175–250
Dye Sublimation Printer	206–400
360 dpi Color Inkjet	150
720 dpi Color Inkjet	240–300
1440 dpi Color Inkjet	240–300

Image file formats

When images are in your computer's memory and are displayed on a screen, they are all basically the same, regardless of where they came from—they differ only in height, width, and bit depth (and, of course, content). More significant differences are introduced when you save the image data to a file on a hard drive, CD-ROM, DVD, floppy disk, or other storage medium. When you save image data, it is stored according to a set of rules describing how the information is organized and encoded. These standardized sets of rules for storing data are called *file formats*.

ImageJ native formats

ImageJ can open and save images in most common image file formats: TIFF (uncompressed), GIF, JPEG, and BMP, as well as the FITS and DICOM formats used by the science community. Free plug-ins can be installed that allow ImageJ to open and save images in other familiar formats, including Photoshop, PNG, PICT, and PCX, as well as QuickTime® and AVI movie formats. (See **What's in a name?** at right for more about these formats.)

Comparing standard formats—TIFF, GIF, and JPEG

The TIFF format is very versatile and can be read by most graphic and image processing applications. TIFF files can contain single or multiple images of a wide range of bit depths. A key feature of the TIFF format is that the file can include a variety of *tags* that record related information such as the spatial scale, time, and geographic location of the image.

- Open the **Image Formats** folder, then the **Comparing Formats** folder, and open the **Scene RGB color.tif** image.

The disadvantage of the TIFF format is that the files tend to be large compared to other formats.

23. Read the memory size of this image from the image window status bar, and record it in Table 1 on your Data Sheet.

Scientifically, TIFF files are superior because they are *lossless*—that is, they preserve all the original image data. The TIFF format specification allows for various forms of compression to produce smaller files, but ImageJ cannot read or write compressed TIFF files. (See **Compressed TIFF files**, at right, for more on this topic.)

You can, however, save TIFF files using lossless ZIP compression by choosing **File > Save As... > ZIP...** The size of a zipped file depends on the content of the image with simpler images compressing more than complex images. This image occupies only 680 KB of storage as a ZIP-compressed file.

- Open the **Scene RGB color.zip** file.

24. Read the memory size of this image from the image window status bar, and record it in Table 1 on your Data Sheet. Compare the quality of the image to the original image, and record the quality in the appropriate column of the table. See **Image quality descriptions** at right.

- Open the **Scene 8-bit color.tif** file.

This is also a TIFF format image, but each pixel requires only one byte, so the saved image requires less space, at 240 KB. It looks very close to the original file, but the quality is reduced slightly—zoom in on the mushrooms and compare the enlarged image with the original.

Compressed TIFF files

TIFF files can be compressed to take up less storage space. This can be done using several methods, but most require licensing of software routines that add to the cost of the software. Since ImageJ is public domain software, there is no practical way to include the compression/decompression routines in the program needed to handle compressed TIFF files.

If you receive compressed TIFF files that you want to use with ImageJ, you may be able to open and re-save them as uncompressed TIFF files using commercial image viewing or editing software such as Adobe Photoshop.

What's in a name?

Here are definitions of some of the common image format names that ImageJ supports:

AVI = Audio-Video Interleaved

BMP = Windows Bitmap

DICOM = Digital Imaging and COmmunications in Medicine

PNG = Portable Network Graphic

FITS = Flexible Information Transport System

GIF = Graphic Interchange Format

JPEG = Joint Photography Experts Group format (also called JFIF).

PCX = PC Paintbrush Exchange

PICT = Macintosh picture format

TIFF = Tagged Image File Format

Image quality descriptions

Use the following key to determine the image quality description to record in the table on your answer sheet:

- **High** - Image is identical or nearly identical in quality to the original.
- **Medium** - Image looks OK but differences are noticeable if you look closely.
- **Low** - Image looks terrible with too few colors and/or serious compression artifacts.

- Close all the image windows except for one copy of the **Scene RGB color.tif** image (the original image).

25. Record the memory size and quality of this image on your Data Sheet.

JPEG format is a *lossy* format that is often used for compressing images for Web pages and e-mail to reduce the time required to transmit the data. The compression, or quality, setting can range from 0 (lowest quality) to 100 (highest quality). JPEG compression attempts to preserve the appearance of the image but changes the pixel values in the process.

- Open **Scene Quality 100.jpg**, **Scene Quality 50.jpg**, and **Scene Quality 10.jpg**, and examine them carefully.

As you increase the compression level (decrease the quality), you can see more compression artifacts, particularly at the edges of objects.

26. Record the memory size and quality of each of the three JPEG images in the table on your Data Sheet.

- Close the three **Scene Quality** images.

GIF format is also used to create small files for use on Web pages and e-mail attachments.

- Open **Scene 256 colors.gif** and **Scene 12 colors.gif**, and examine them carefully.

27. Record the memory size and quality of the two GIF images in the table on your Data Sheet.

- Close all the open image windows.

Before you decide which image format is “best,” let’s look at an image with very different content and saved in each format we have just explored.

- Open the **Graphic RGB color.tif** file.

This image will be our original file. Keep it open as you examine and record information about the other **Graphic** files.

- Open the other 7 **Graphic** image files. Examine each image carefully as you open it.

28. Record the memory size and quality of each image file in the table on your Data Sheet.

- Close all the open image windows.

More about JPEG Compression

- In the **JPEG** folder, open all the images in the **Flowers** folder.

These images were saved with JPEG quality settings ranging from 100 (highest quality) to 0 (lowest quality). The quality setting is included in the file name of each image.

29. For this scene, what is the minimum quality level that you think produces a “high quality” image?

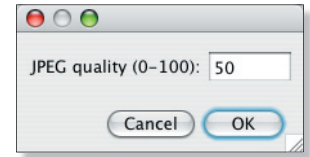
- Close all the **Flowers** images, then open all the images in the **Wheels** folder.

30. Do you think JPEG is a good format for this type of image? Why?

- Close all the **Wheels** images.

JPEG Quality

In **ImageJ**, you can set the **JPEG Quality** setting to any value from 0 to 100 by choosing **Edit > Options > JPEG Quality...**



More about GIF images

- 🖼 In the **GIF** folder, open the **Flowers** folder, then open all the images in the **Dithered** folder.

These images were saved in GIF format with a specific number of colors, ranging from 256 (most colors) to 2 (fewest colors). The number of colors is included in the file name of each image. ImageJ does not produce dithered GIF images, so these were created using Adobe Photoshop.

- 🖼 Tile the images, and compare them.

31. For this scene, what is the minimum quality level that you think produces a “high quality” image?

- 🖼 Close all the flower images, then open all the images in the **Not Dithered** folder.

These images were saved using ImageJ by converting the original RGB Color image to 8-bit color, using the specified number of colors.

32. What do you think about using GIF format for this type of scene?

Choosing image formats

The key to choosing the “right” file format is knowing what you plan to do with the image. Most often, the option for choosing the file format comes when the digitizing software prompts you to save the image. You can change the bit depth or convert between color and grayscale in ImageJ by choosing **Image > Type**, or change the file format by using the **Save As...** option.

Use the table of data you recorded on your Data Sheet to answer the following questions about image formats. Feel free to open and re-examine any of the image files to help you answer the questions.

33. Does the memory size of an open image depend on the file format the image was saved in? If not, explain what *does* determine the memory size of an open image.
34. Which image format saves a graphic image in the smallest file without sacrificing image quality?
35. Which image format saves a continuous tone (“scenic”) image in the smallest file without sacrificing image quality?
36. If you wanted to scan some plant leaves on your scanner to measure area and brightness differences, what file format would you want to use? Explain why you chose this format.
37. If you wanted to save a labeled diagram consisting of lines, shapes, and text that you created in ImageJ to put on a Web page, which file format would you use? Explain why you chose this format.
38. If you needed to add an image taken of the plant that the leaves came from to the same Web page, which file format would you use? Explain why you chose this format.
39. If you don’t ever intend to measure pixel values in an image, what would be the advantage of saving the image in JPEG format?
40. If storage space is not an issue, what bit depth, spatial resolution, and file format would you choose for saving an image file for later use? Explain why you chose these specifications.

Dithering defined

Dithering is a method of simulating colors or shades of gray that aren’t in the image by altering the values of adjacent pixels to give the illusion of an intermediate color.

On your own

- Use a scanner to digitize a photograph at varying resolutions, and compare the quality. Scan an image at 150 dpi and 300 dpi, then scale the 300 dpi image by a factor of 0.5 (50%) and compare it to the 150 dpi image. Is there a difference?
- Convert 24-bit RGB color images taken with a digital camera to 8-bit color format. Magnify and compare the images. Try sharpening the images or reducing noise.
- The **Formats** folder includes folders containing DICOM and FITS images. Open several of these images using ImageJ, then choose **Image > Show Info...** These image formats include headers, which provide a wealth of information about the images.
- ImageJ provides limited support for digitizing video from cameras, VCRs, DVDs, etc. Refer to the ImageJ Web site and documentation for additional information.



Digitizing Details

1. Describe how the appearance of the image changes as the bit depth decreases from 8 bits to 1 bit.
2. At what bit depth does the image quality begin to degrade noticeably?
3. Complete the table showing the number of possible colors that correspond to the following bit depths.

Bit depth	Expanded Representation	Exponential Representation	Possible Colors
1	2	2^1	2
2	2×2	2^2	4
3	$2 \times 2 \times 2$	2^3	
4	$2 \times 2 \times 2 \times 2$	2^4	
5	$2 \times 2 \times 2 \times 2 \times 2$	2^5	
6	$2 \times 2 \times 2 \times 2 \times 2 \times 2$	2^6	
7	$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$	2^7	
8	$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$	2^8	

4. Based on the table, what is the maximum number of colors that the **Sue (4-bit).tif** image can contain?
5. According to its histogram, how many different pixel values, or colors, does the 4-bit image contain? (Count the number of lines, or “spikes,” in the histogram. Don’t forget to count the spikes at each end of the histogram.)
6. According to the image window status bar, what is the size of the **Balloon 8-bit.tif** image, in kilobytes (K)? How well does this match your calculated value?
7. This image has the same width and height as the 8-bit version. Without peeking, predict how much memory you think this image will occupy.

8. Compare your prediction with the actual memory size reported in the image window.

9. Predict the size, in kilobytes, of this image, then compare it to the actual memory size reported in the image window.

10. What is an advantage of 32-bit floating point image data?

11. Assuming that this image has the same width and height as the other balloon images, predict its size in kilobytes. Compare your prediction with the actual memory size reported in the image window.

12. Calculate the total amount of memory required for this image in *megabytes*. (To convert from kilobytes to megabytes, you must divide the size in kilobytes by 1024!) Compare your prediction to the actual memory size reported in the image window status bar.

13. What happens to the quality of the image when you convert it to indexed color? In which part(s) of the image are the colors most poorly reproduced?

14. In general, what happens to amount of memory required for an image as the bit depth increases?

15. What happens to the size of an image as its resolution increases?

16. What are the dimensions of the 300 dpi image?

17. Calculate how much memory this image would require. Remember, memory size equals:

$$\frac{\text{Width (px)} \times \text{Height (px)} \times \text{Bit depth (bytes/px)} \times \text{Color Channels}}{1024 \text{ bytes/kilobyte}}$$

18. What happens to the memory size of the image as you double the spatial resolution?



19. Predict the size of this image if you increased the resolution to 1200 dpi.
20. How do the details in the image change as the resolution increases?
21. If you scan an image at 100 dpi and the file size is 50K, what will the file size be if you increase the scan resolution to 300 dpi?
22. Using the output device resolution table, what would be the ideal image resolution for printing on a 600 dpi laser printer?

23–28.

Image Name	Open Size (KB)	Saved Size (KB)	Quality (High / Medium / Low)
Scene RGB Color.tif		712 KB	
Scene RGB Color.zip		680 KB	
Scene 8-bit Color.tif		240 KB	
Scene 256 Colors.gif		256 KB	
Scene 12 Colors.gif		84 KB	
Scene Quality 100.jpg		328 KB	
Scene Quality 50.jpg		56 KB	
Scene Quality 10.jpg		20 KB	
Graphic RGB Color.tif		712 KB	
Graphic RGB Color.zip		8 KB	
Graphic 8-bit Color.tif		240 KB	
Graphic 256 Colors.gif		8 KB	
Graphic 12 Colors.gif		4 KB	
Graphic Quality 100.jpg		36 KB	
Graphic Quality 50.jpg		12 KB	
Graphic Quality 10.jpg		8 KB	

29. For this scene, what is the minimum quality level that you think produces a “high quality” image?
30. Do you think JPEG is an appropriate format for this type of image? Explain your answer.
31. For this scene, what is the minimum quality level that you think produces a “high quality” image?
32. What do you think about using GIF format for this type of scene?

33. Does the memory size of an open image depend on the file format the image was saved in? If not, explain what *does* determine the memory size of an open image.

34. Which image format saves a graphic image in the smallest file without sacrificing image quality?

35. Which image format saves a continuous tone (“scenic”) image in the smallest file without sacrificing image quality?

36. If you wanted to scan some plant leaves on your scanner to measure area and brightness differences, what file format would you want to use? Explain why you chose this format.

37. If you wanted to save a labeled diagram consisting of lines, shapes, and text that you created in ImageJ to put on a Web page, which file format would you use? Explain why you chose this format.

38. If you needed to add an image taken of the plant that the leaves came from to the same Web page, which file format would you use? Explain why you chose this format.

39. If you don’t ever intend to measure pixel values in an image, what would be the advantage of saving the image in JPEG format?

40. If storage space is not an issue, what bit depth, spatial resolution, and file format would you choose for saving an image file for later use? Explain why you chose these specifications.

