1495 UNIVERSITY OF ABERDEEN | CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Image Mining – 1

Data Mining & Visualisation

Lecture 29

2025

Today...

- Digital Images
- Image Enhancement and Preprocessing

Image Mining

As smartphones, digital cameras, CCTV, and embedded image sensors become increasingly ubiquitous, the volume of image data being captured and stored continues to grow rapidly.

As a result, the tools, techniques, and academic disciplines concerning the analysis of image data are also continuing to develop and evolve.

Image Mining

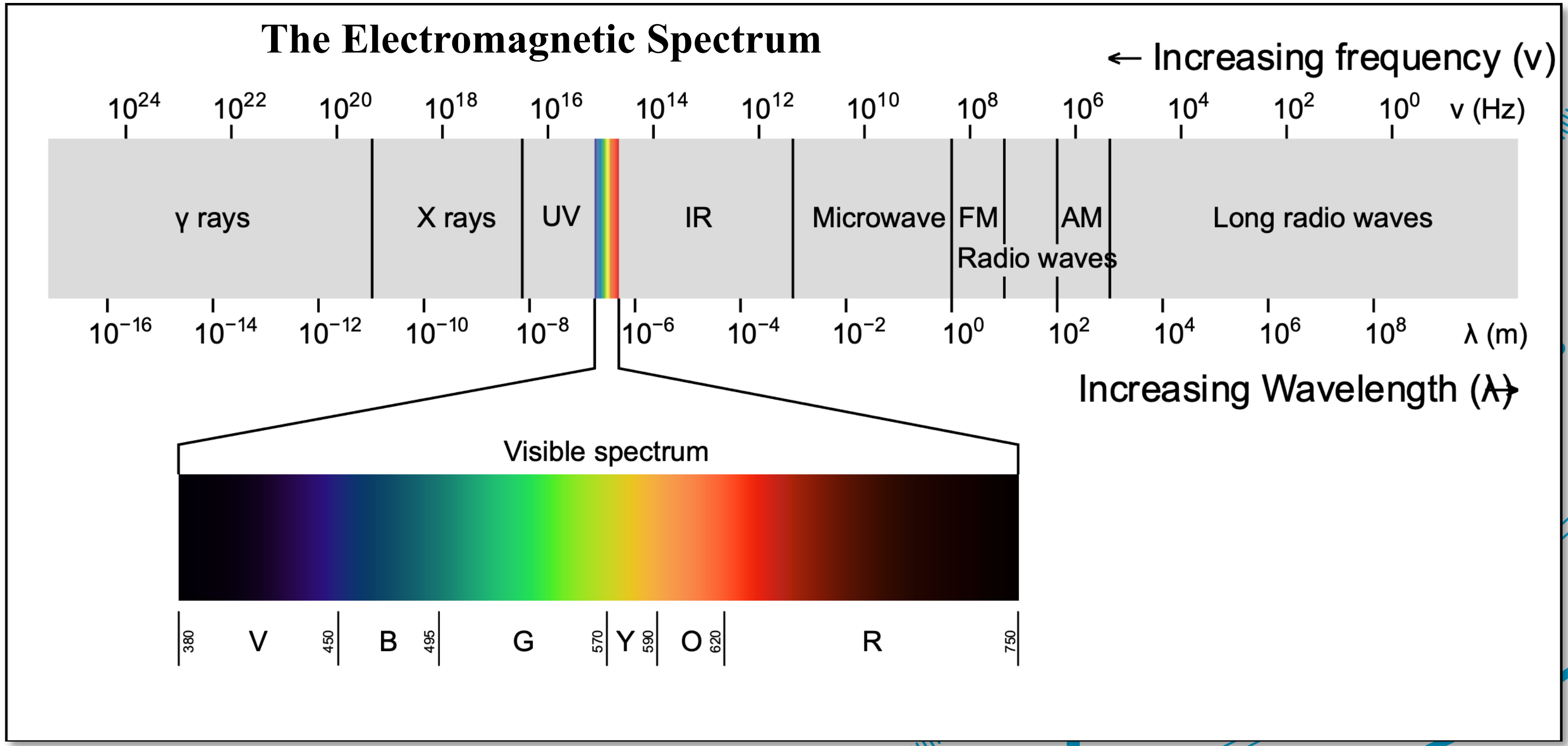
We’re going to discuss some of the key aspects of how digital images can be represented, processed and analysed.

To start off, it’s useful to have a basic understanding of how light, the human perception system, and digital cameras work.

Digital Images

The Electromagnetic Spectrum

What we think of as (visible) light is a small part of a spectrum of electromagnetic energy, containing radio waves, microwaves, infrared and ultraviolet light, and X and gamma rays.



The Electromagnetic Spectrum

Within that visible spectrum, the wavelength determines how we perceive it.

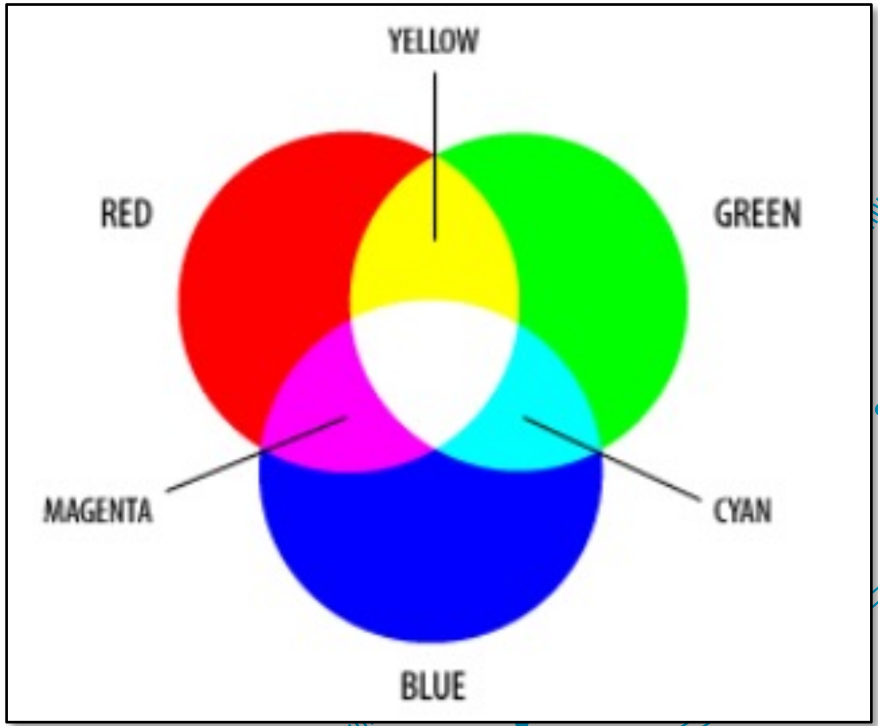
We see:

~ 700 nm wavelength light as red

~ 550 nm wavelength light as green

~ 450 nm wavelength light as blue

Different combinations of these wavelengths result in different colours (e.g. red + green = yellow).

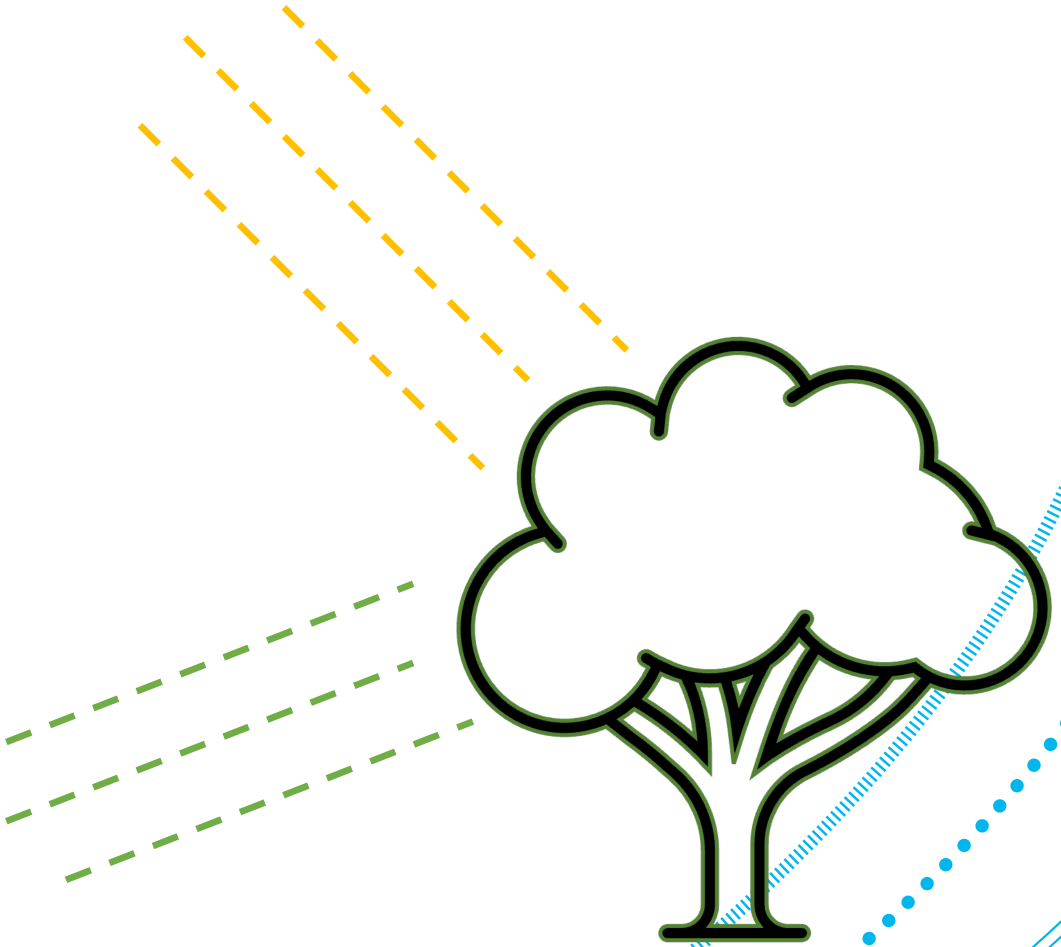


Light Reflection

When we 'see' an object, what we are actually seeing is light reflecting off of that object.

The object absorbs certain wavelengths of light and reflects other wavelengths.

Those reflected wavelengths define what we see and what colour(s) they appear to be.



Human Perception System

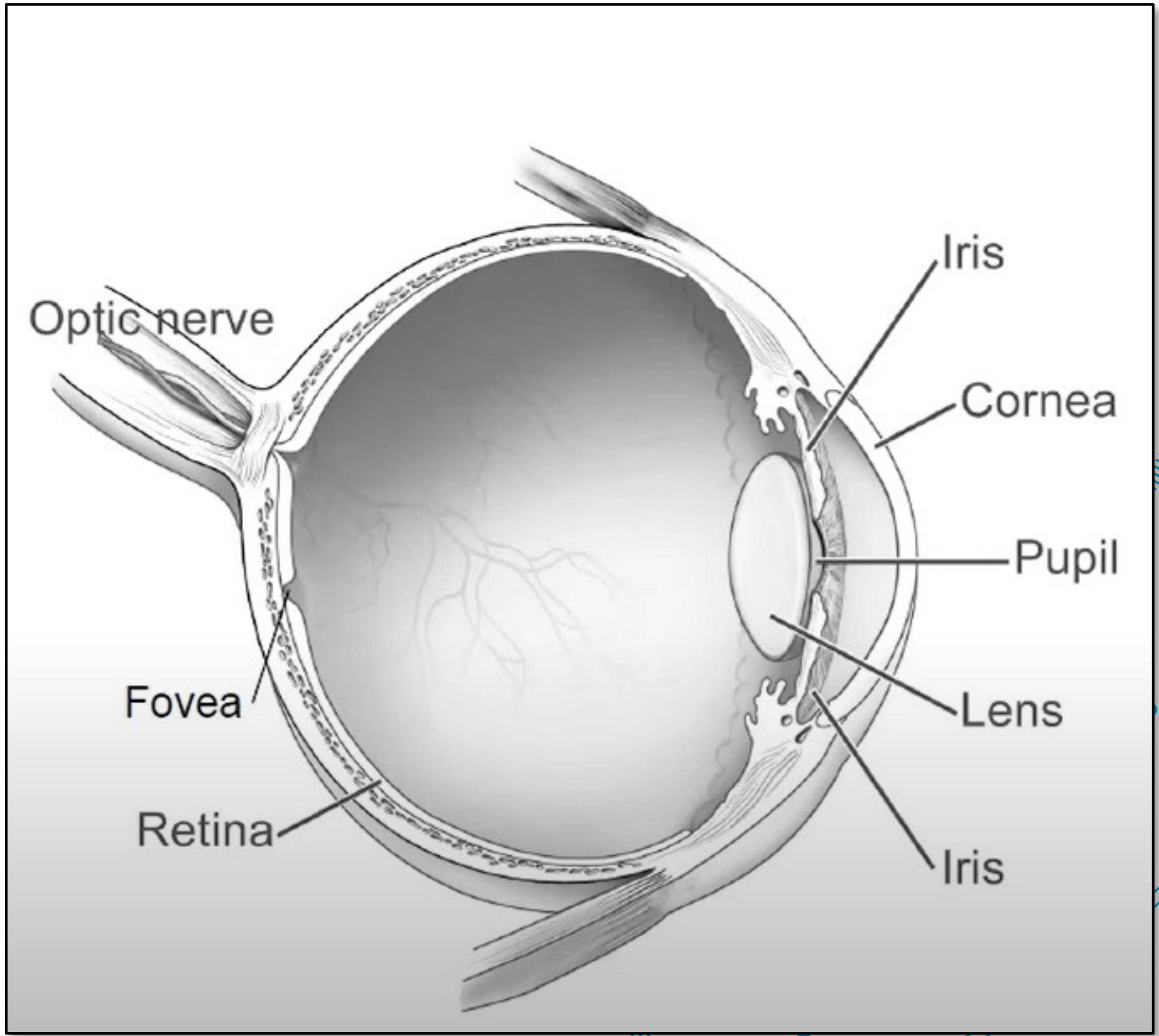
Light enters the **pupil**.

The **iris** contracts and expands, controlling how much light enters the eye.

The **lens** focuses the light onto the retina, where light sensitive cells are.

The **fovea** is where the highest concentration of light sensitive cells are.

The **optic nerve** carries all of these signals to the brain, for processing.



Digital Cameras

The way that digital cameras work is broadly inspired (to a degree) by the human perception system.

They have an array of light sensitive sensors (called photosites).

The camera's shutter opens, exposing this array to a stream of light particles.

When the shutter closes, the camera measures the strength of electric signal for each photosite, corresponding to the amount of light that it was exposed to.

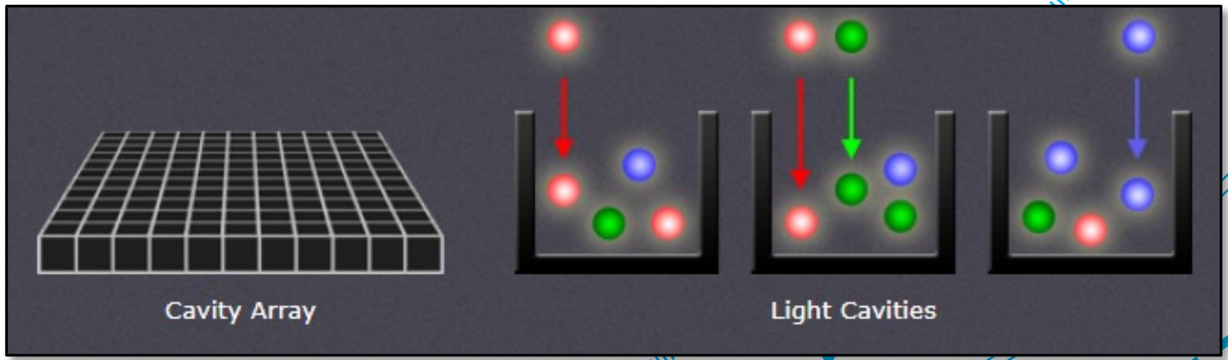
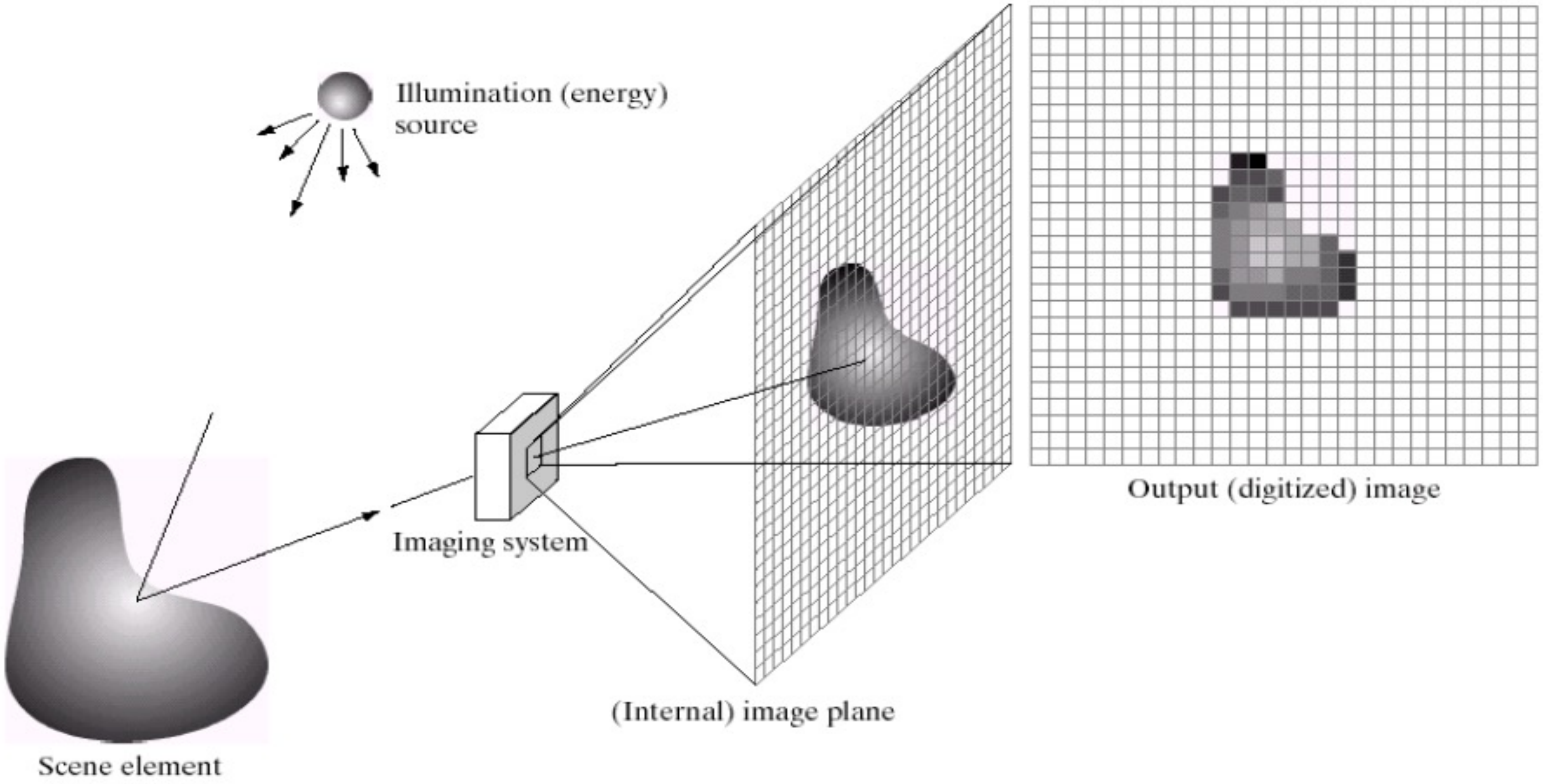


Image Acquisition and Representation



An example of the digital image acquisition process. (a) Energy source, (b) An element of a scene. (c) Imaging system. (d) projection of the scene onto the image plane. (e) digitalized image.

Digital Cameras

More accurately, each of these photosites will typically have a colour filter applied.

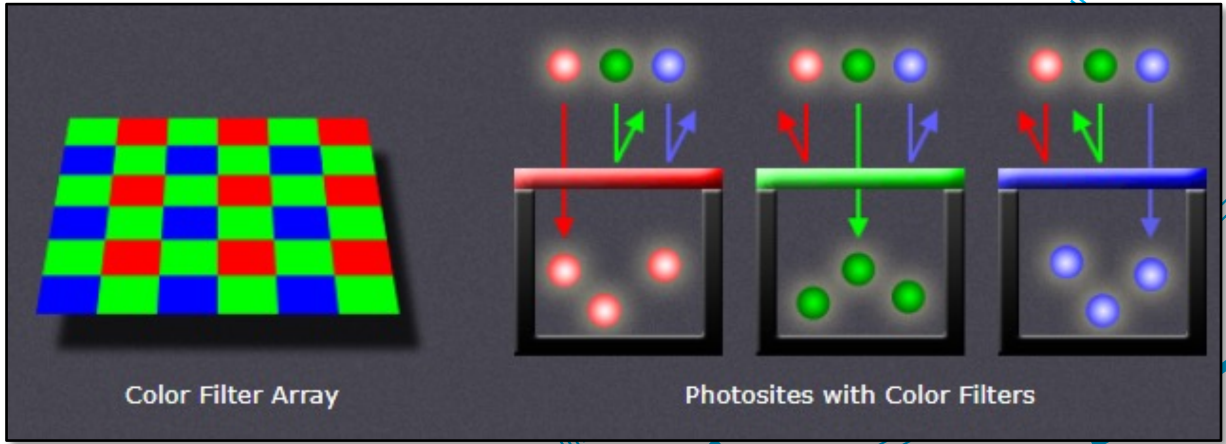
These filters only allow certain light wavelengths to pass, so that only red light is absorbed by a photosite with a red filter, green light by a green filter, etc.

These colour filtered photosites are typically arranged in a series of repeating 2x2 patterns, containing:

1x red filtered photosite,

1x blue filtered photosite, and

2x green filtered photosites.

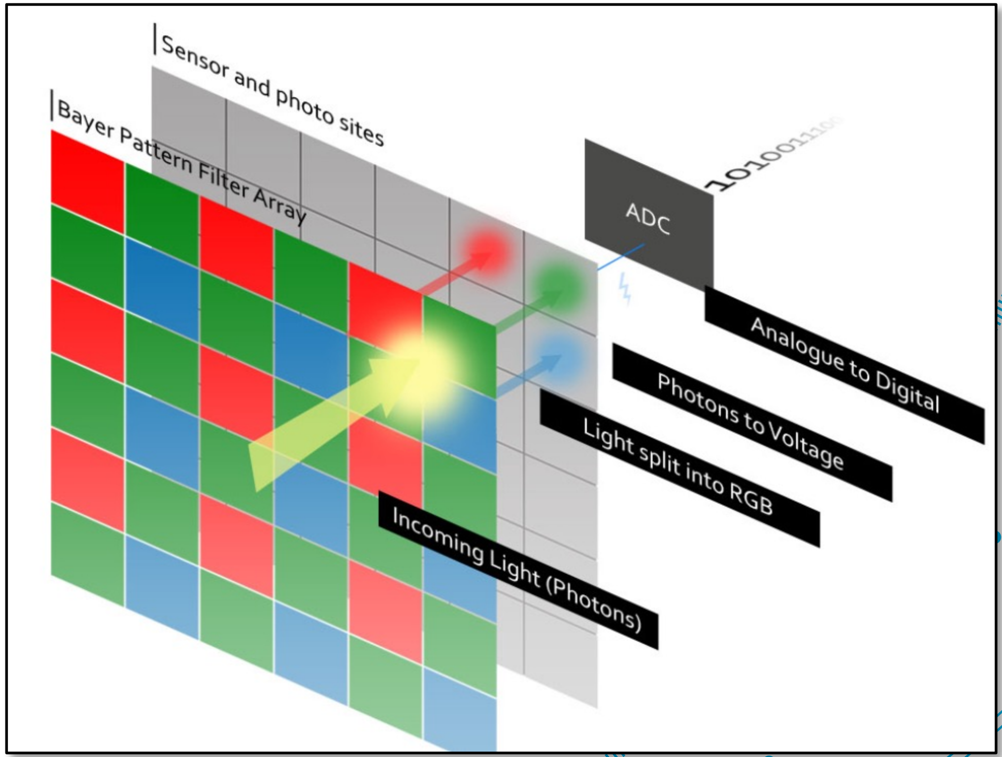


Digital Cameras

This 2x2 pattern is called a Bayer array.

After the camera's shutter is closed, each photosite's electric signal is converted into an 8-bit digital value.

These individual 8-bit values are then processed together to generate RGB pixel values.



Demosaicing

The digital values for each of these 2x2 patterns are used to reconstruct full RGB values for each pixel, using interpolation.

This process is known as **demosaicing**.

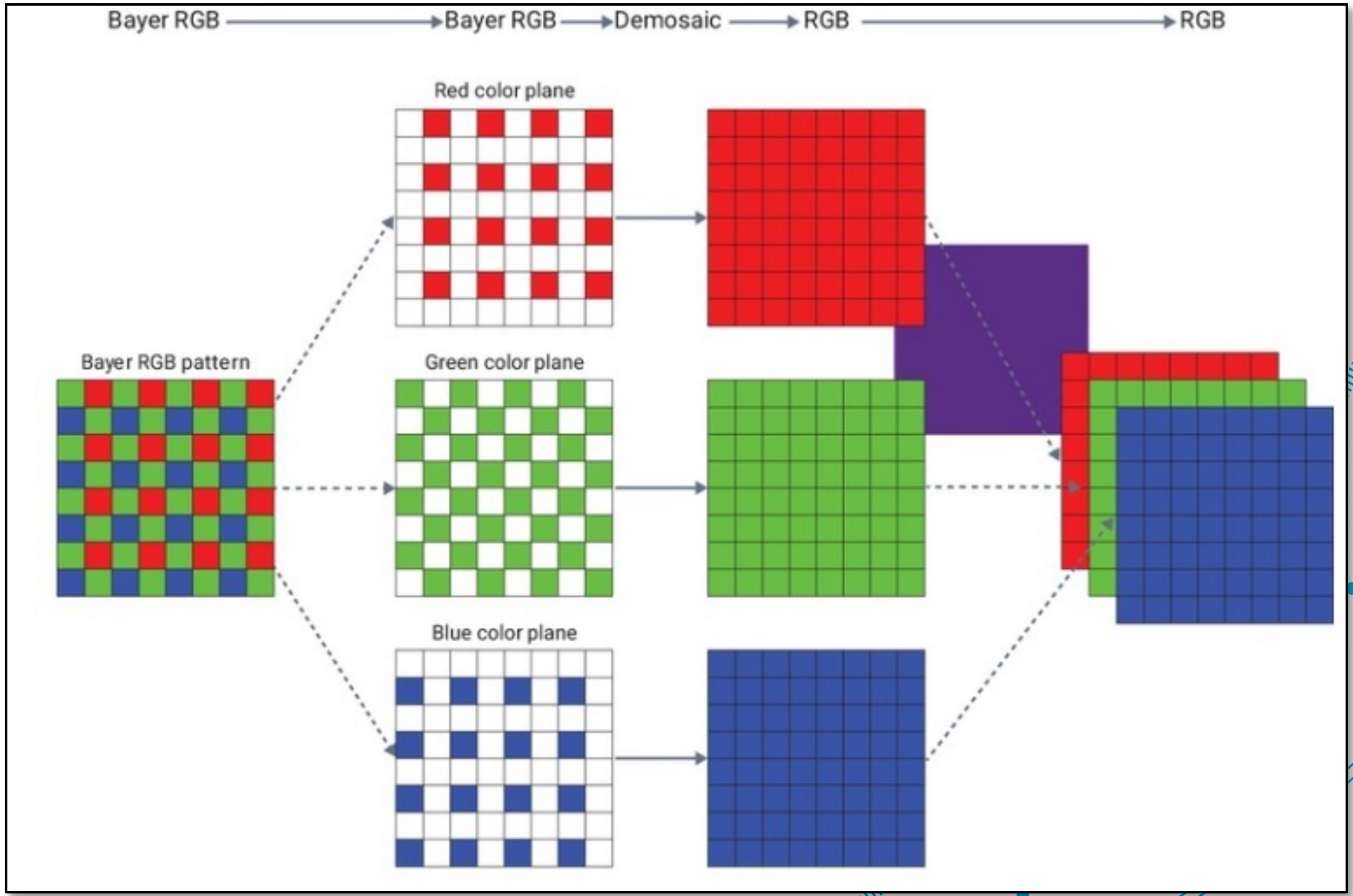
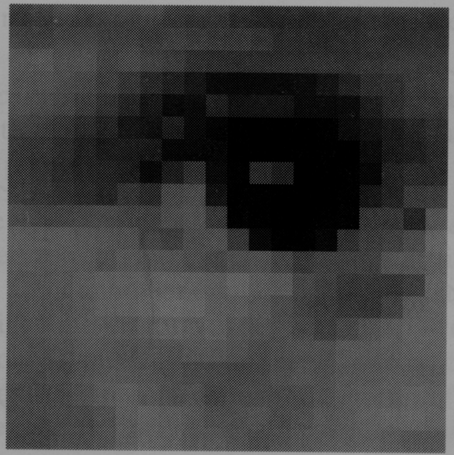


Image Representation

Digital images are represented by a 2-dimensional grid of pixel values: P[x, y]

The overall size of the image is x*y pixels.



117	125	133	127	130	130	133	121	116	115	100	91	93	94	99	103	112	105	109	106
134	133	138	138	132	134	130	133	128	123	121	113	106	102	99	106	113	109	109	113
146	147	138	140	125	134	124	115	102	96	93	94	99	96	99	100	103	110	109	110
144	141	136	130	120	108	88	74	53	37	31	37	35	39	53	79	93	100	109	116
139	136	129	119	102	85	58	31	41	77	51	53	53	33	37	41	69	94	105	108
132	127	117	102	87	57	49	77	42	28	17	15	13	13	17	41	53	69	88	100
124	120	108	94	72	74	72	31	35	31	15	13	15	11	15	13	46	75	83	96
125	115	102	93	88	82	42	79	113	41	19	100	82	11	11	17	31	91	99	100
124	116	109	99	91	113	99	140	144	57	20	20	15	11	15	17	63	87	119	124
136	133	133	135	138	133	132	144	150	120	24	17	15	15	17	20	115	113	88	150
158	157	157	154	149	145	133	127	146	150	116	35	20	19	28	105	124	128	141	171
155	154	156	155	146	155	154	154	147	139	148	150	138	120	128	129	130	151	156	165
150	151	154	162	166	167	169	174	172	167	177	166	164	140	134	120	121	120	127	172
145	149	151	157	165	169	173	179	176	166	166	157	145	136	129	124	120	136	163	168
144	148	153	160	159	158	165	172	165	169	157	151	149	141	130	140	151	162	169	167
144	141	147	155	154	149	156	151	157	157	151	144	147	147	149	159	158	159	166	165
139	140	140	150	153	151	150	146	140	139	138	140	145	151	149	156	156	162	162	161
136	134	138	146	156	164	153	146	145	136	139	139	140	141	149	157	159	161	169	166
136	133	136	135	144	159	168	159	151	142	141	145	139	146	153	156	164	167	172	168
133	129	140	142	146	159	167	165	154	151	146	141	147	154	156	160	161	157	153	154

Image types

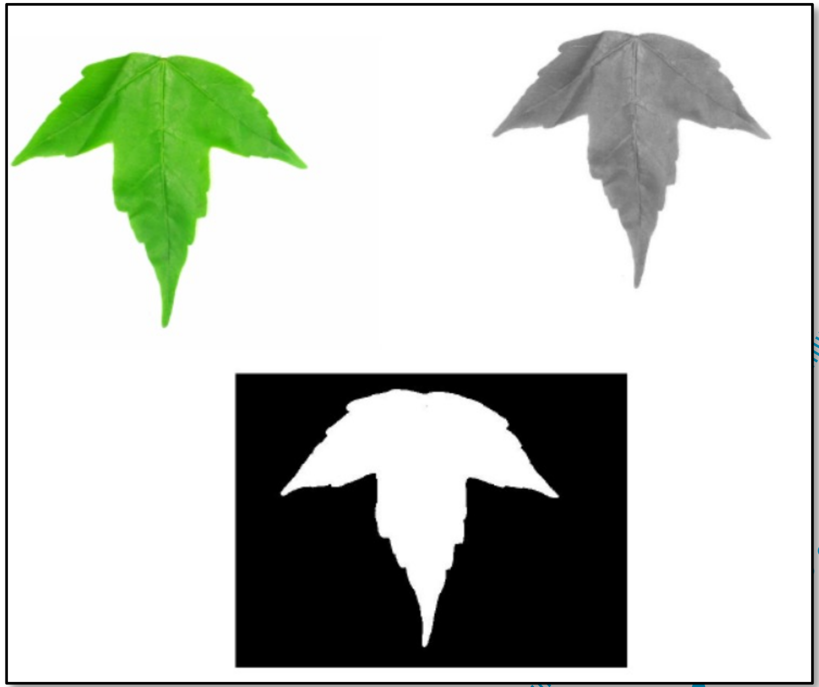
Pixels can be represented in different ways, which will change the ‘type’ of image.

This includes:

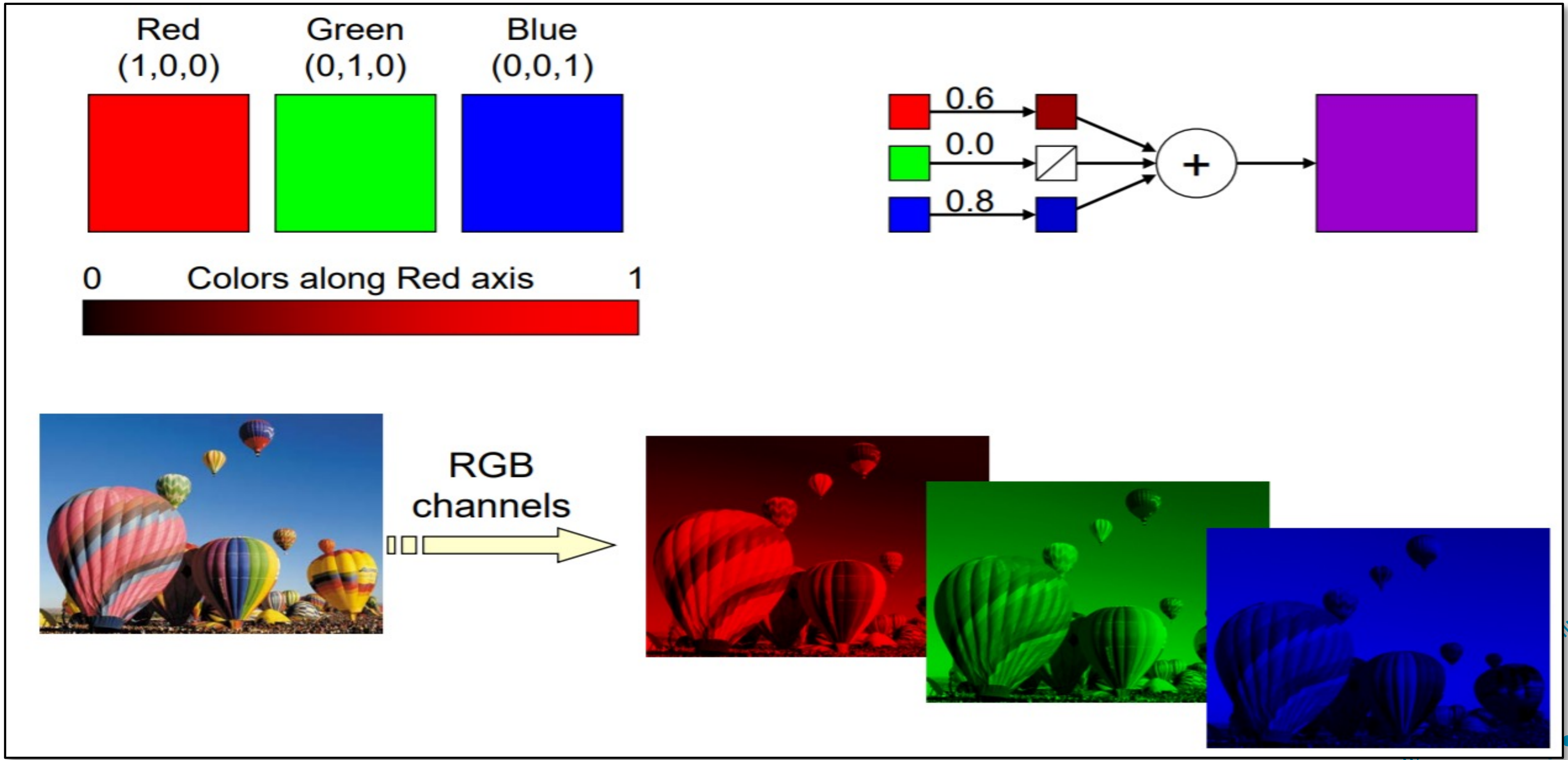
RGB images: Three 8-bit channels; R, G, and B

Grayscale images: One 8-bit channel

Binary images: One 1-bit channel (true or false)



RGB Images



Red
(1,0,0)
Green
(0,1,0)
Blue
(0,0,1)
0.6
0.0
0.8
+
0
Colors along Red axis
1
RGB
channels

Colour Terms

RGB pixels can be described in terms of their:

Intensity / brightness / value: i.e. how much light the pixel contains

Hue: the dominant colour

Saturation: the purity or vividness of the colour

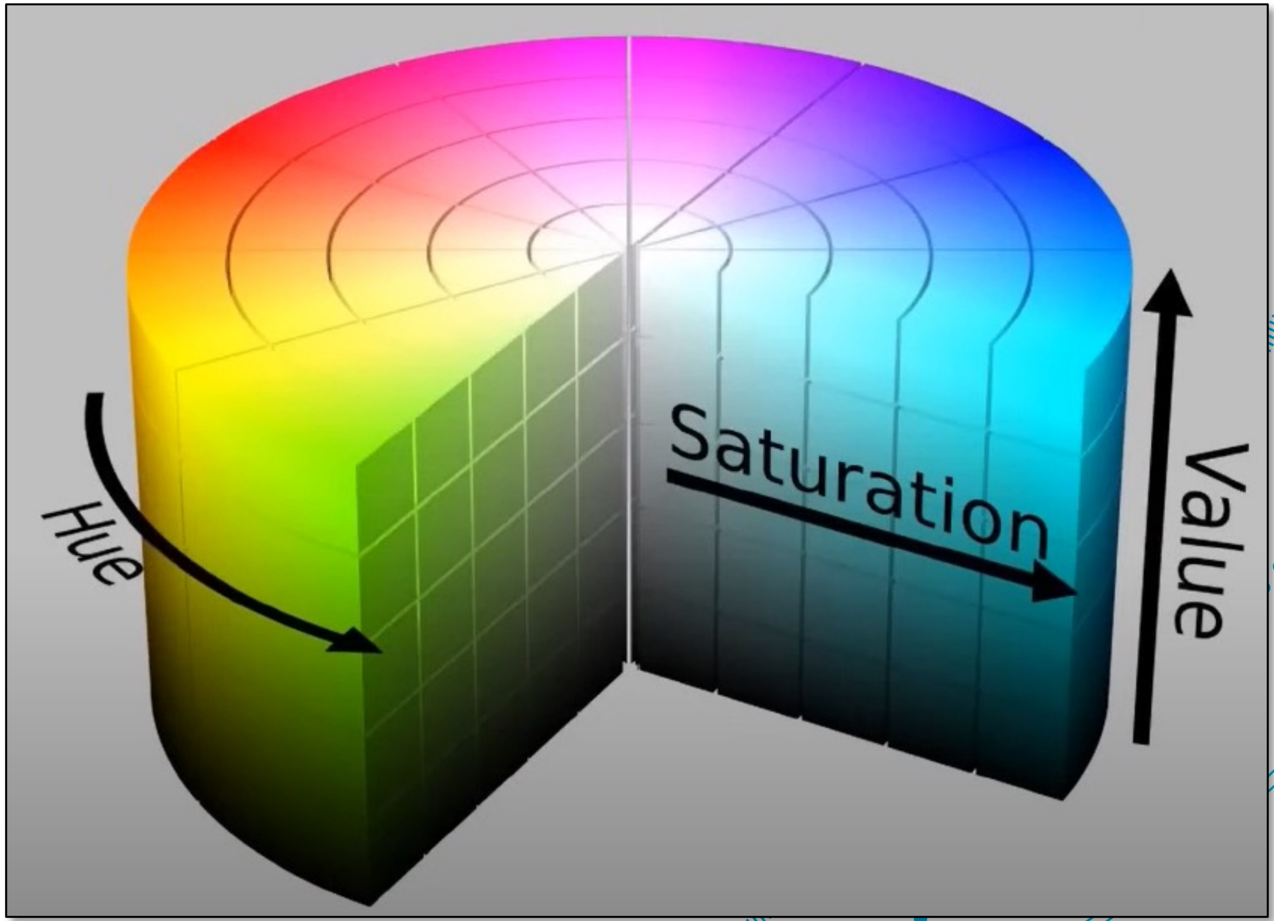


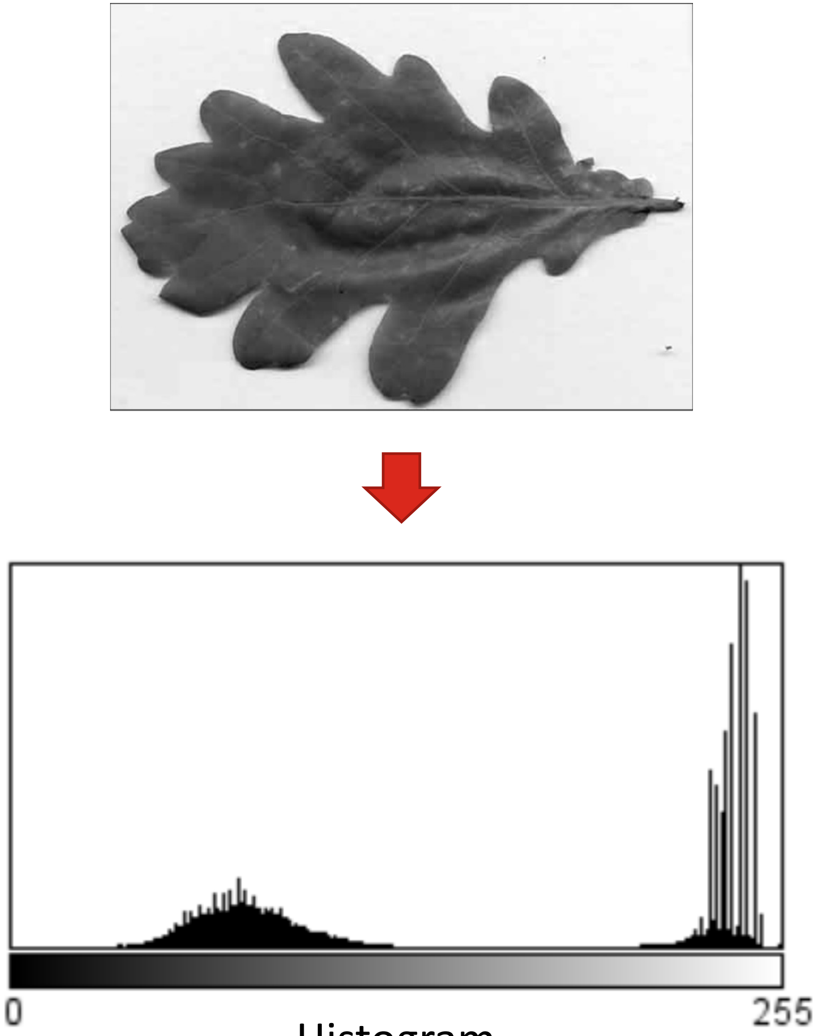
Image Histogram

Image histograms show us the frequency distribution of intensity values within the image.

They plot the number of pixels corresponding to each pixel value.

By looking at the histogram, we can judge the entire tonal distribution of the original image at a glance.

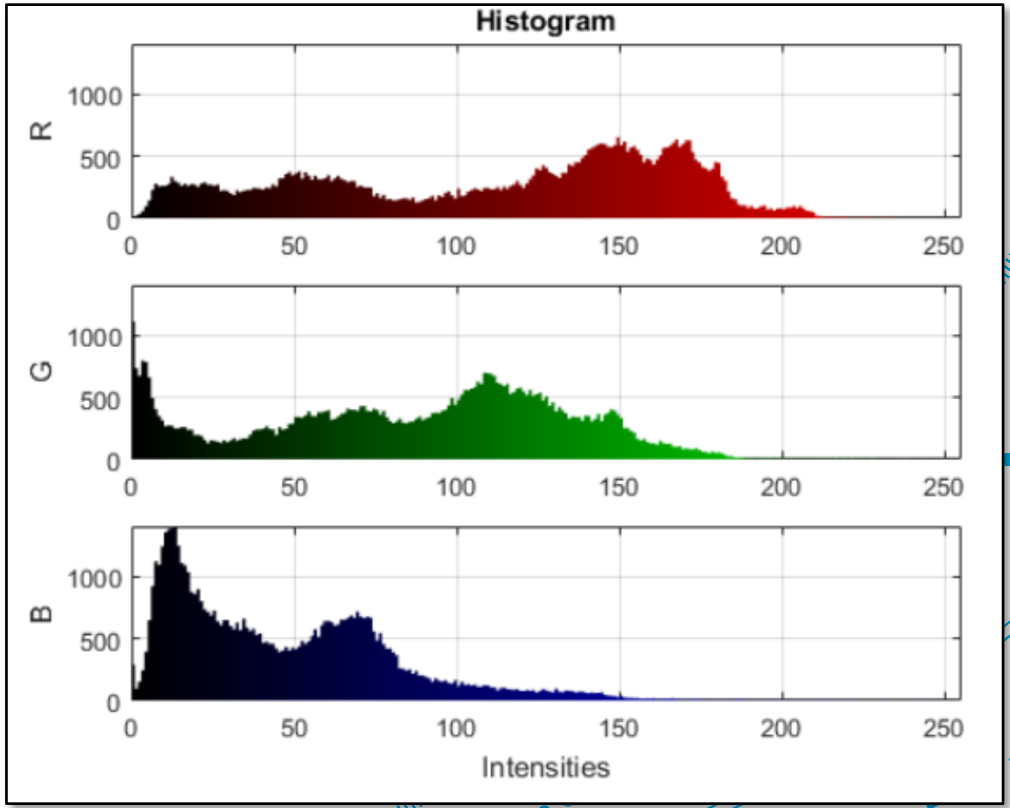
Grayscale image



Histogram

Image Histogram

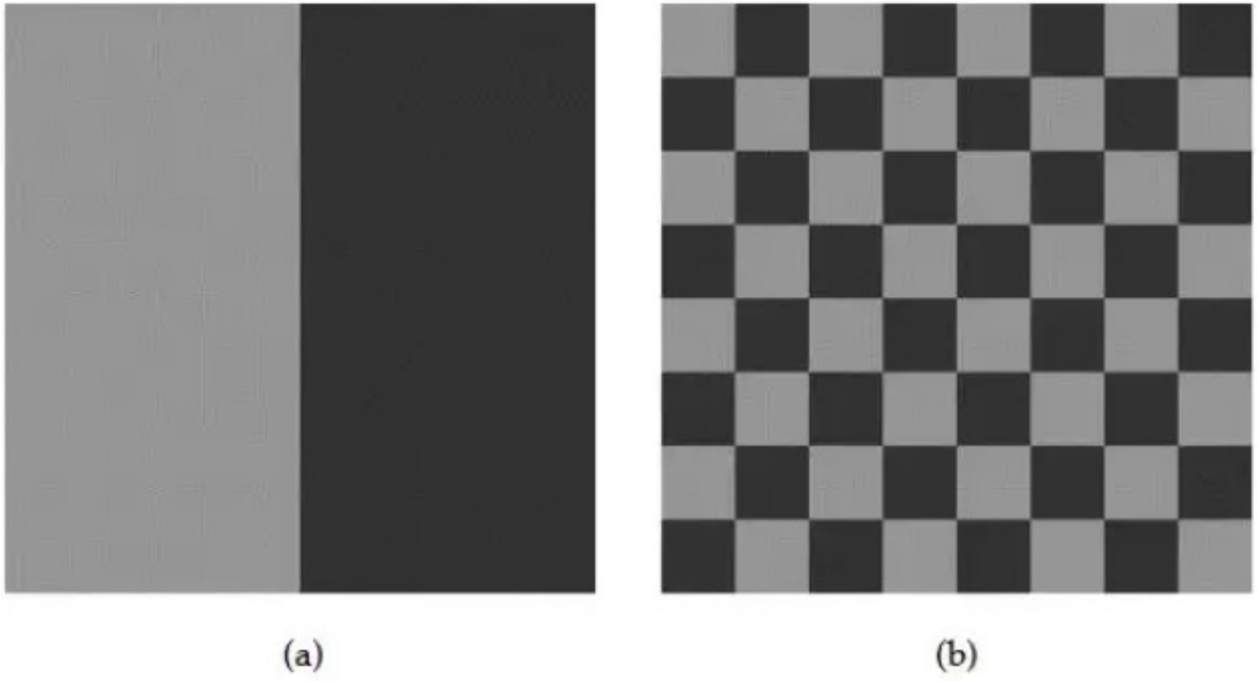
For RGB images, the idea is the same, but we can also have a separate histogram representing each colour channel.



Histogram

Importantly, histograms only provide statistical information about the distribution of colour values.

They do not retain any of the *spatial* information about the picture itself.



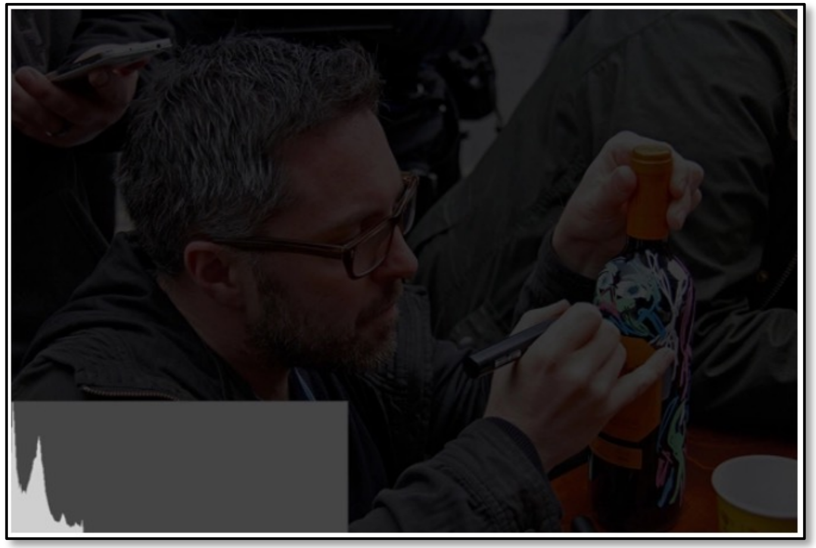
Two different pictures, a. and b., provide the same histogram due to the equal number of the intensities

Image Quality

However, they can give us information about the image itself (i.e. whether it is under- or over-exposed.

Original

Too dark (under-exposed)



Too light (over-exposed)

Image Quality

What we're often looking for in an image is having a balanced histogram.

I.e., showing a bell-shaped curve.

Image Enhancement and Preprocessing

Image Enhancement (Point Operation)

Point Operation is the term given to modifying the pixel values of an image, without changing in the size, geometry or local structure of that image.

The new value of any given pixel depends only on that pixel's previous value.

Image Enhancement (Point Operation)

Point operations might include:

- Modifying the brightness or contrast of an image
- Clamping (restricting the range) of pixel values
- Inverting the image
- ‘Thresholding’ the image

Image Enhancement – Brightness

For example, if we wished to adjust the brightness of our image, we could simply add (or subtract) some fixed intensity value to each pixel.

Pixel values would then be ‘clamped’, to restrict the pixel values to the full intensity range (i.e. 0-255 for 8-bit images).

Image Enhancement – Contrast Stretching

To adjust the contrast of an image, we can map the lowest and highest intensity values of a particular image to the *min* and *max* values of the full intensity range.

$$f_{ac}(a) = a_{\min} + (a - a_{\text{low}}) \cdot \frac{a_{\max} - a_{\min}}{a_{\text{high}} - a_{\text{low}}}$$

For an 8 bit image:

$$f_{ac}(a) = (a - a_{\text{low}}) \cdot \frac{255}{a_{\text{high}} - a_{\text{low}}}$$

Where:
a is a pixel intensity value,
a_{low} is the **lowest** intensity value of the image, and
a_{high} is the **highest** intensity value of the image

Image Enhancement – Inversion

Inversion is the process of inverting an image, such that pixel values become ‘flipped’.

Pixels with a high intensity value become low, and pixels with a low intensity value become high.

Image Enhancement – Inversion

For each pixel, we simply subtract the pixel's intensity value from the max intensity value.

I.e. for 8-bit images:

inversion(a) = 255 - a

Original Image

0	255	127
10	250	50
200	100	50

Negative Image

255	0	128
245	5	205
55	155	205

8-bit& (3 X 3) Image

Original Image

Negative Image

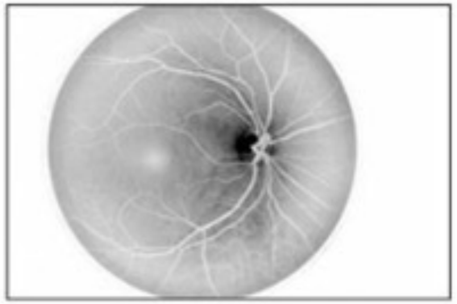


Image Enhancement – Thresholding

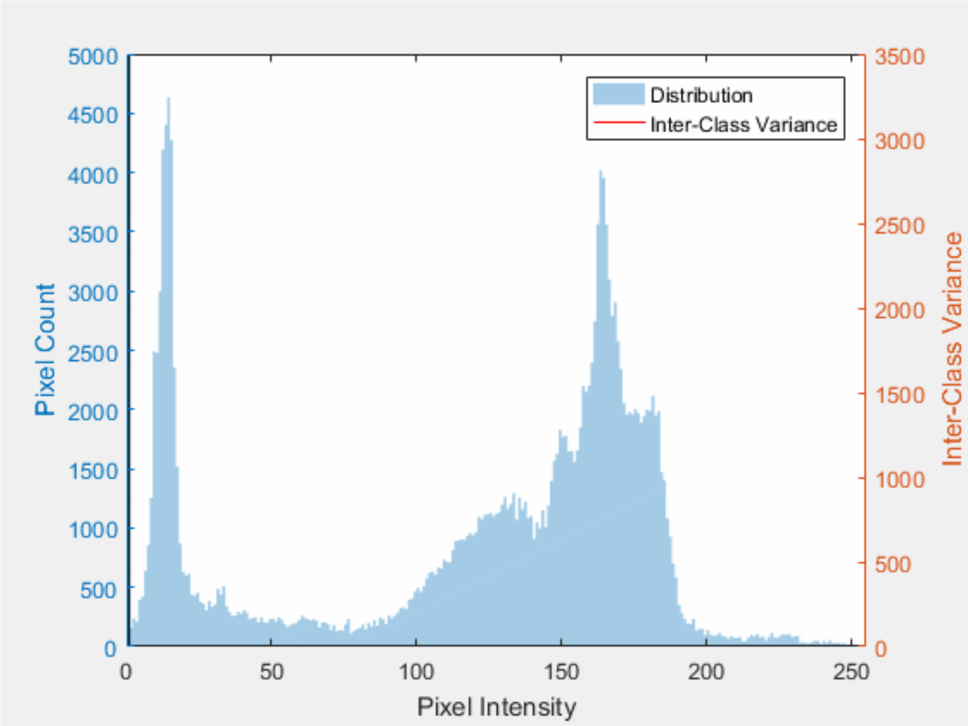
Thresholding is the process of separating the foreground pixels from the background pixels.

It results in the conversion of grayscale or RGB images into binary images.

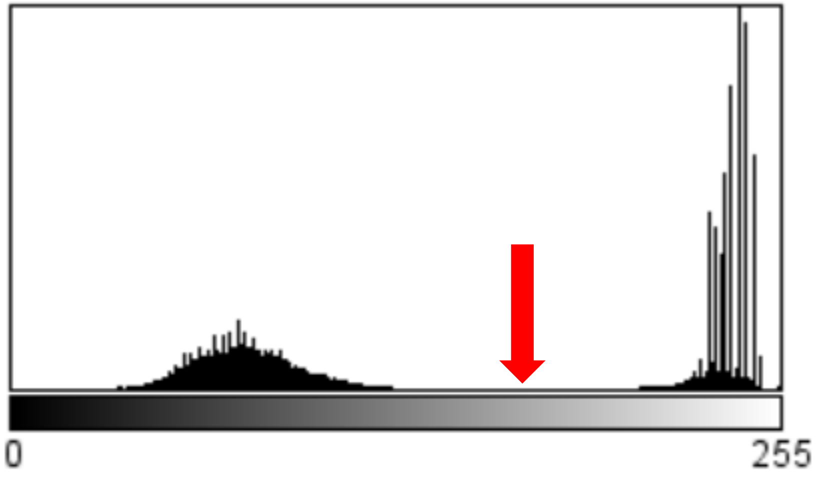
Otsu Thresholding

One example is **'Otsu Thresholding'**, which is a variance-based thresholding technique.

It finds the threshold value which **maximises** the **inter-class variance** between the foreground and background pixels.

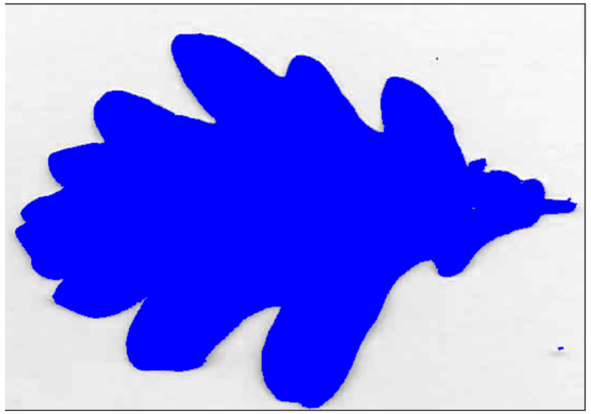


Otsu Thresholding



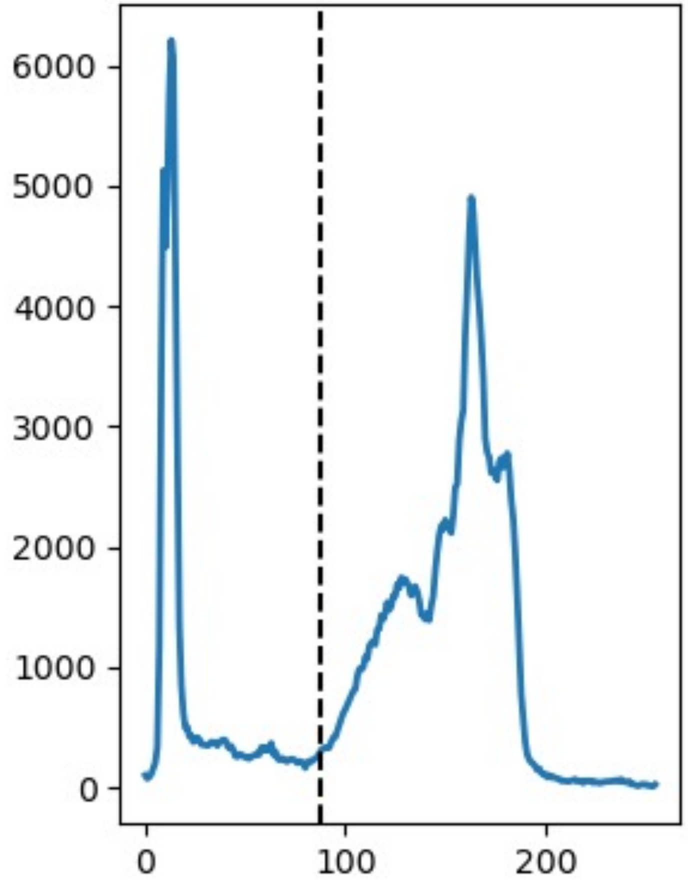
RGB image
(three bands)

Grayscale image
(one band)



binary image (0
and 1)

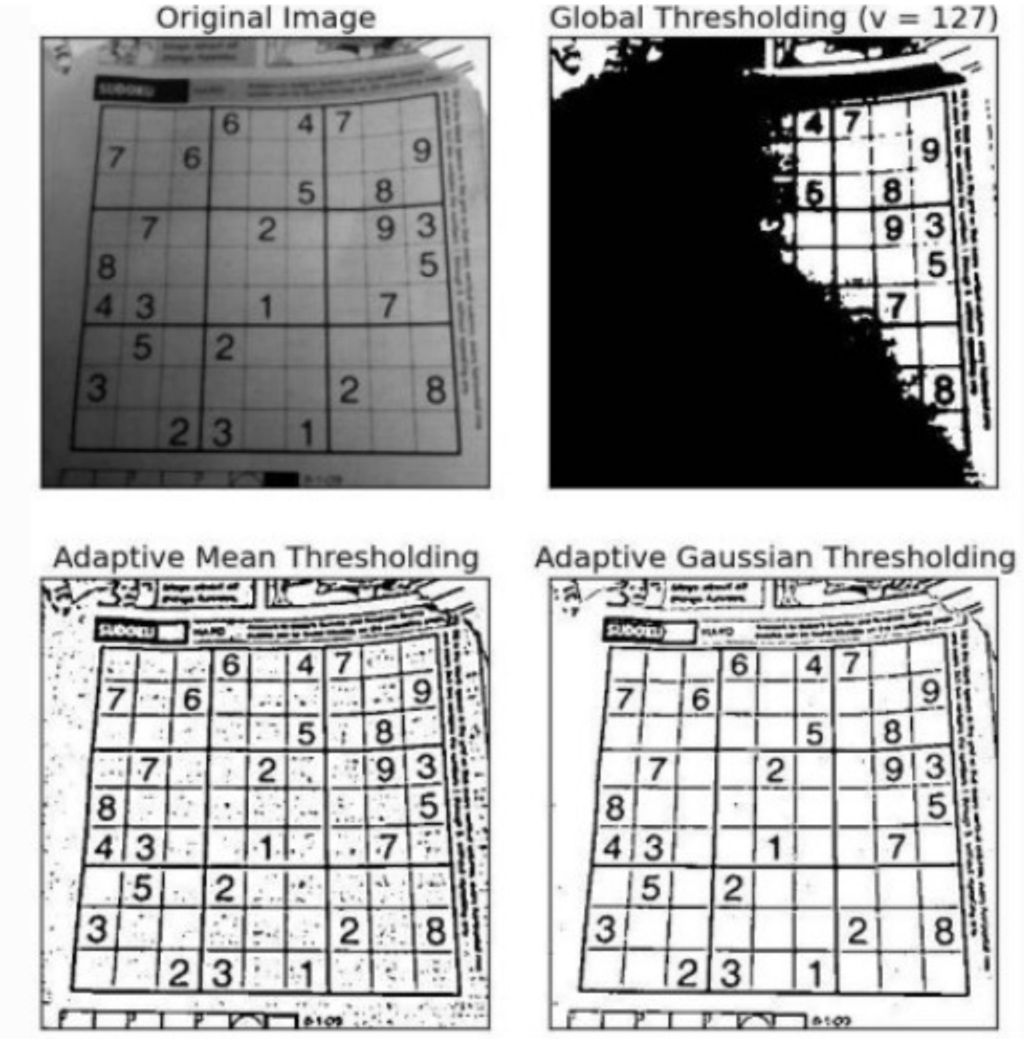
Otsu Thresholding



AdaptiveThresholding

Adaptive Thresholding is another thresholding technique, which works better on images with varying brightness levels.

This approach allows us to get **different threshold values for different regions** of the image.



Mathematical Morphology

Mathematical Morphology

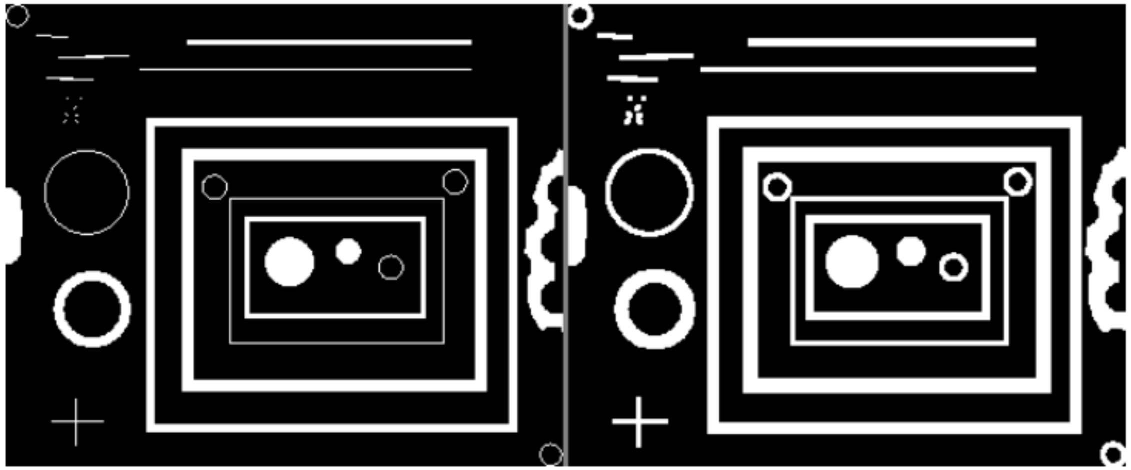
Mathematical morphology is a way of processing the structure and shape of objects in digital images, particularly binary and grayscale images.

It can be used for shape extraction, boundary detection, noise removal, and image segmentation.

Mathematical Morphology

Binary mathematical morphology has two basic operations:

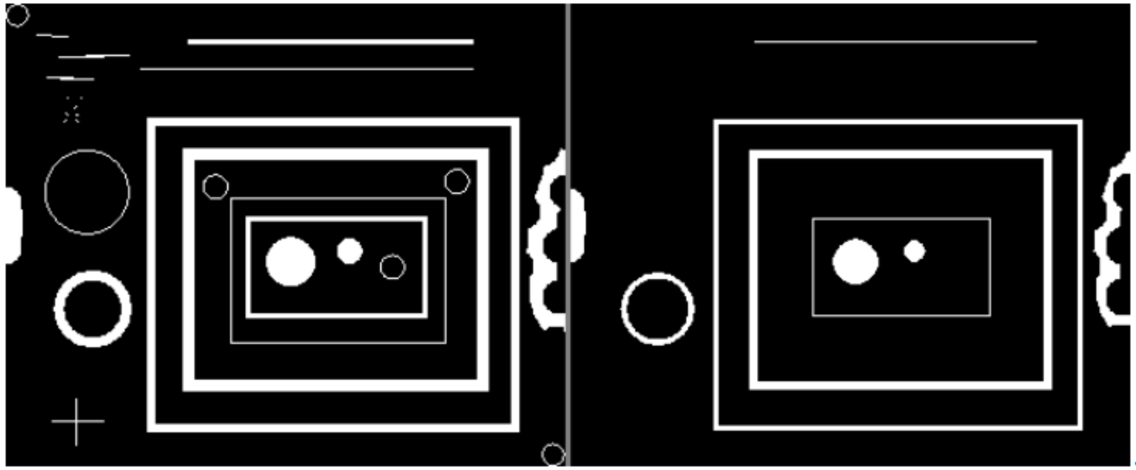
Dilation: making objects more visible and fills in small holes in objects. Lines appear thicker, and filled shapes appear larger.



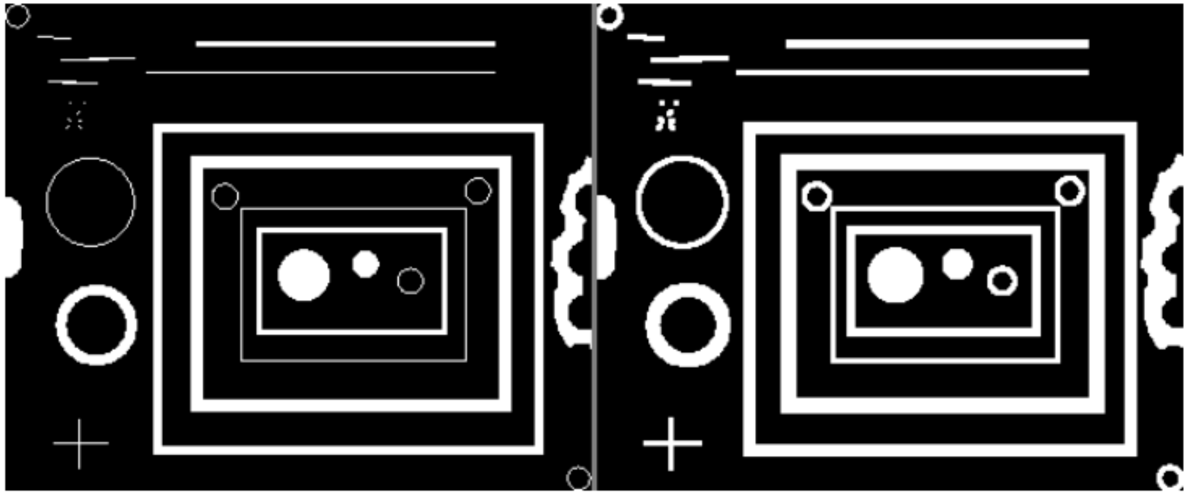
Mathematical Morphology

Binary mathematical morphology has two basic operations:

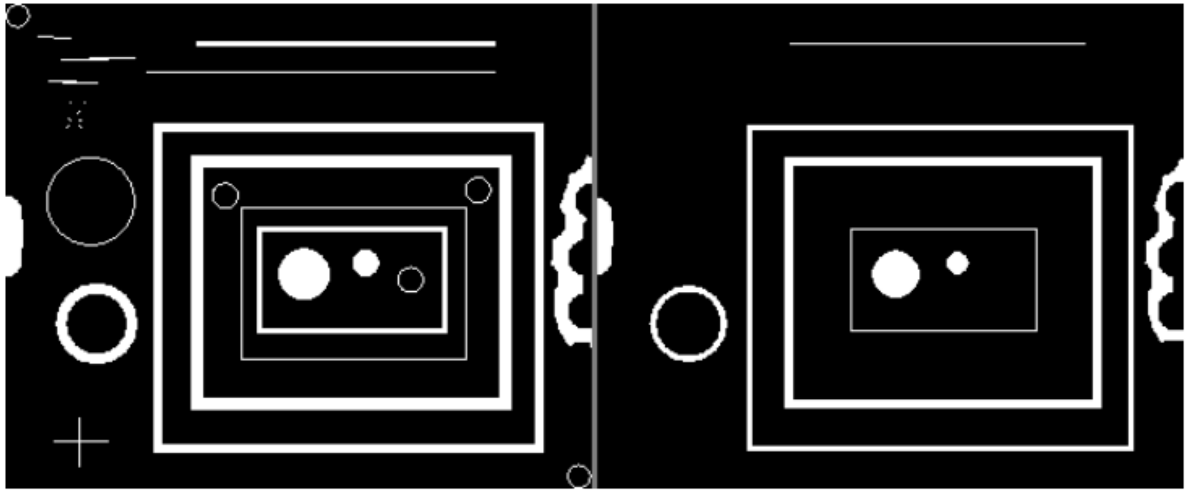
Erosion: Removing floating pixels and thin lines so that only substantive objects remain. Remaining lines appear thinner and shapes appear smaller.



Mathematical Morphology



Dilation

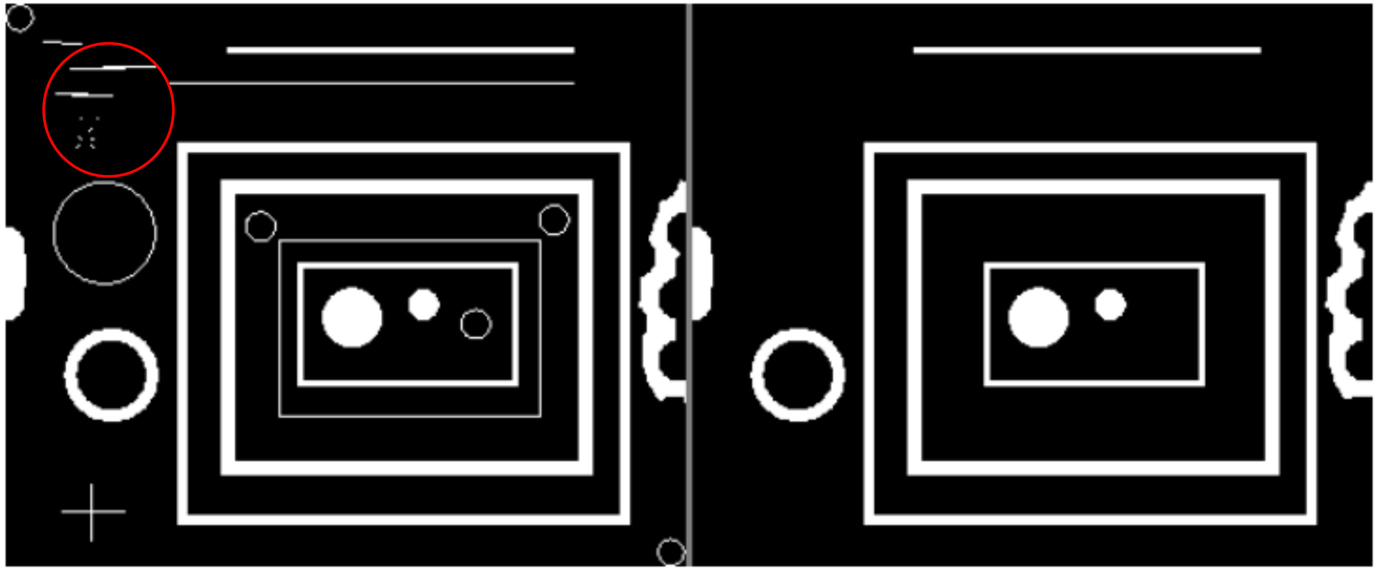


Erosion

Mathematical Opening

The **opening** operation erodes an image and then dilates the eroded image, using the same structing element for both operations.

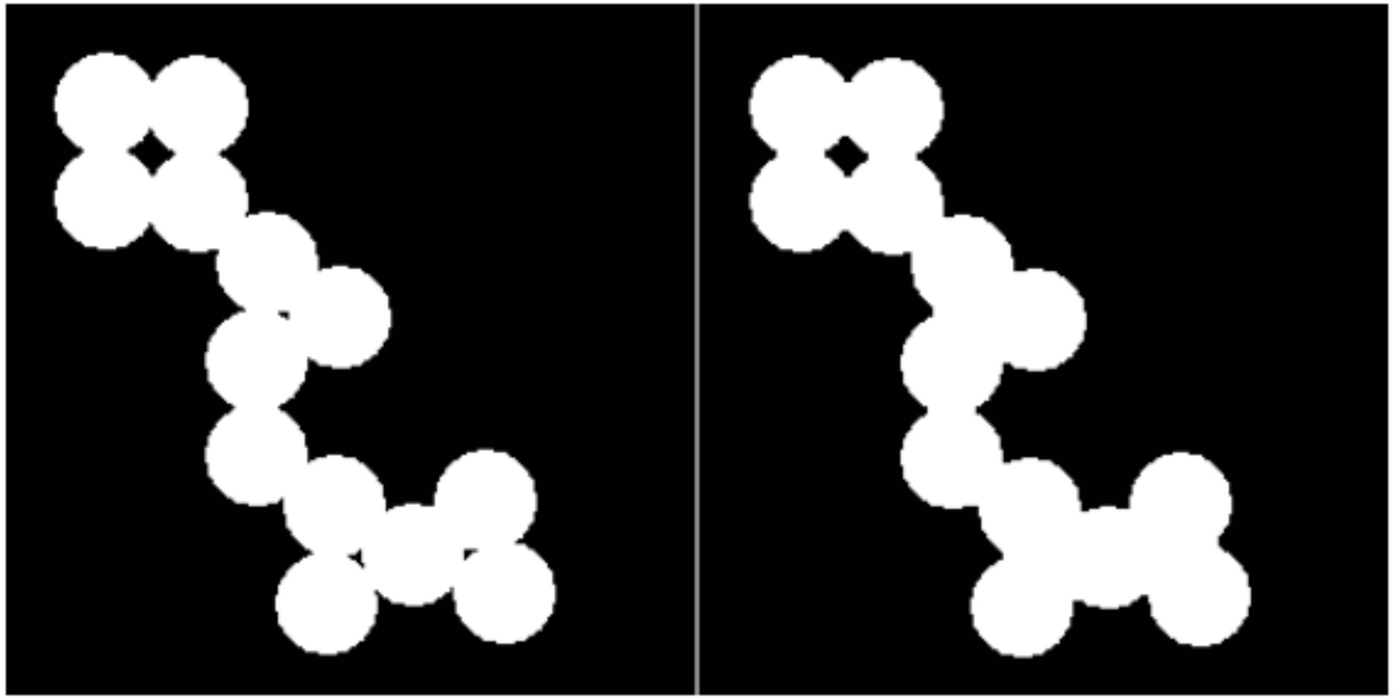
This removes small objects and thin lines from an image, while preserving the shape and size of larger objects.



Mathematical Closing

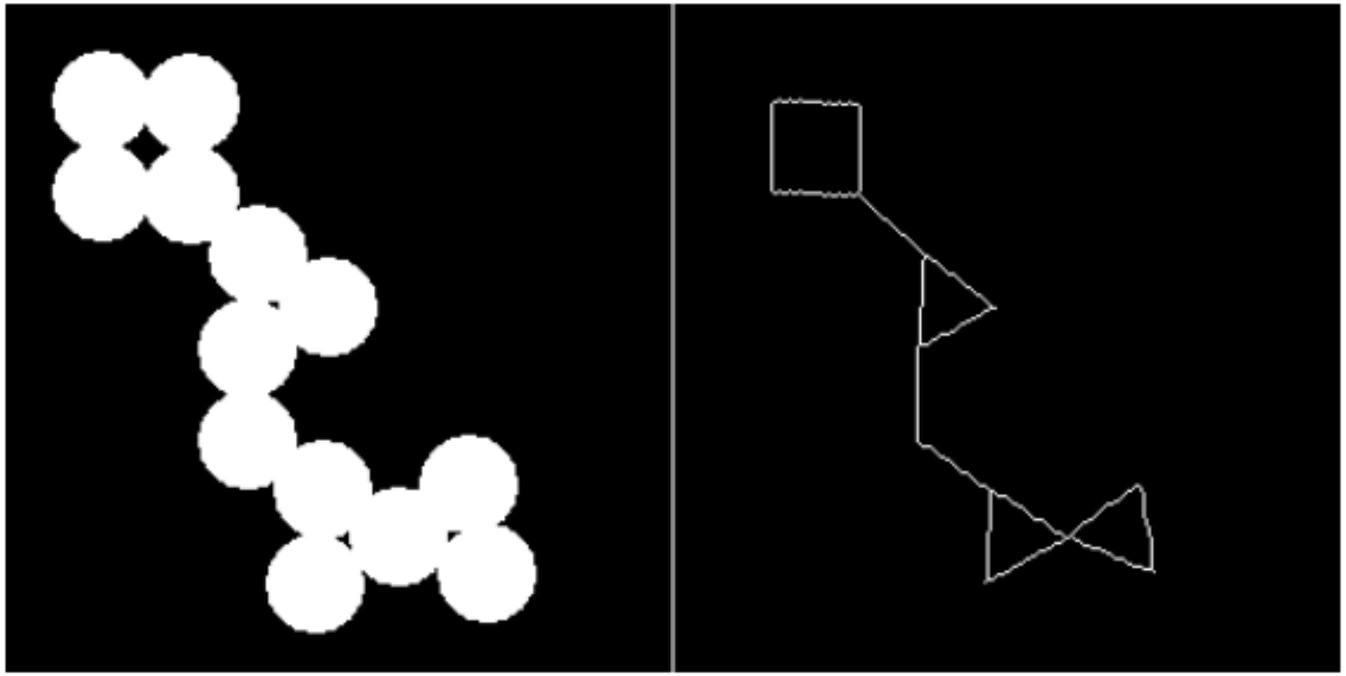
The **closing** operation dilates an image and then erodes the dilated image, using the same structing element for both operations.

This results in filling small holes in an image, while preserving the shape and size of large holes and objects.



Skeletonization

The process of **skeletonization** erodes all objects to centerlines without changing the essential structure of the objects, such as the existence of holes and branches.



Skeletonization

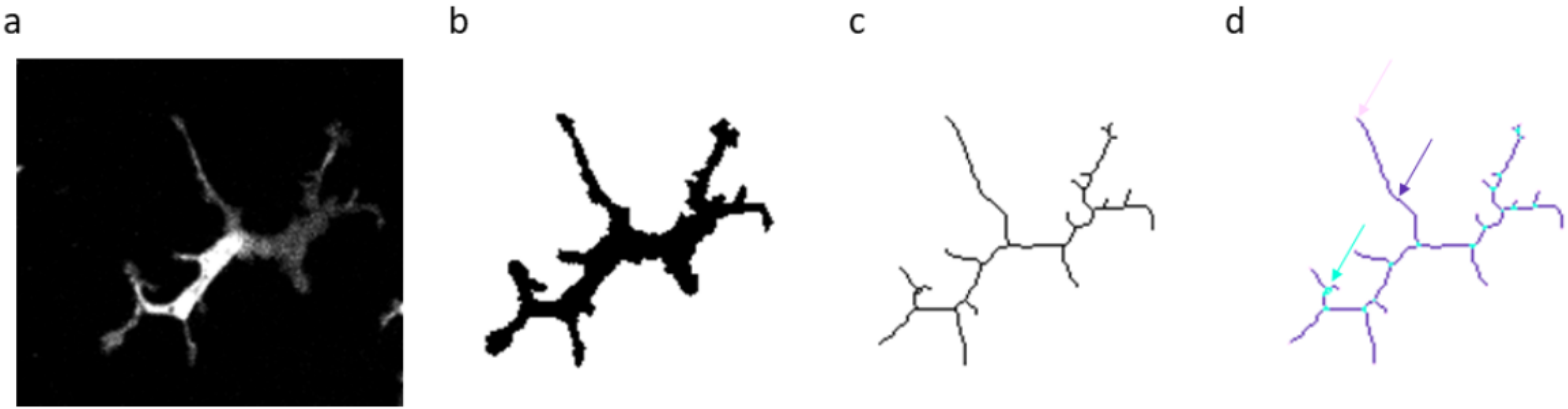


Image before and after skeletonization. a) raw image, b) binary image, c) skeleton image, d) tagged skeleton showing slab pixels (dark purple), junction pixels (cyan), and end-point pixels (pink). Examples of different skeleton pixels are indicated by arrows in the corresponding colors.