Representing images

All **data** needs to be stored as numbers inside computers

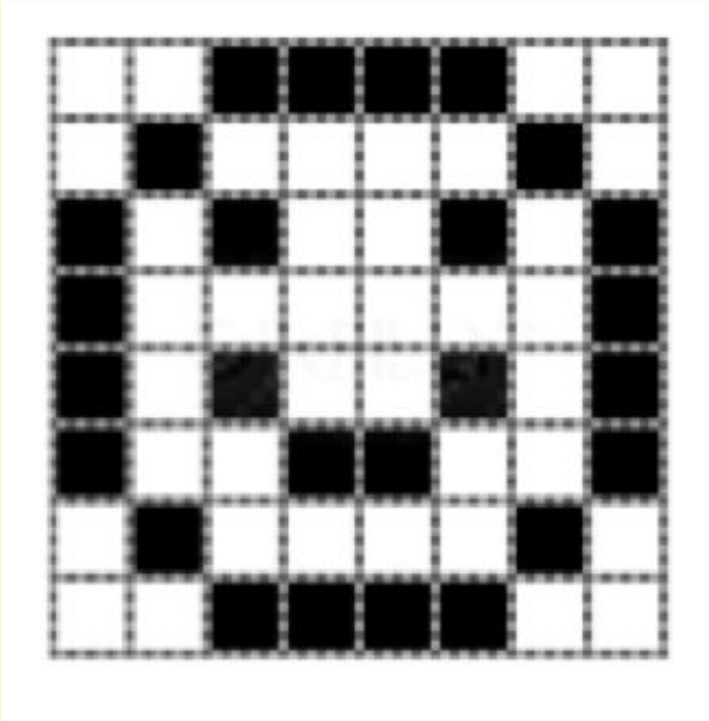
This means that there needs to be a way to **encode** images as numbers

If everyone uses the same code it becomes a **standard**

Images are split into a grid of **pixels**

This allows each pixel to be encoded using a numeric value

A pixel is the smallest part of an image - a **Picture Element**



Black and white bitmaps have two possible colours per pixel

So each pixel takes 1 binary bit to represent it

0 = black

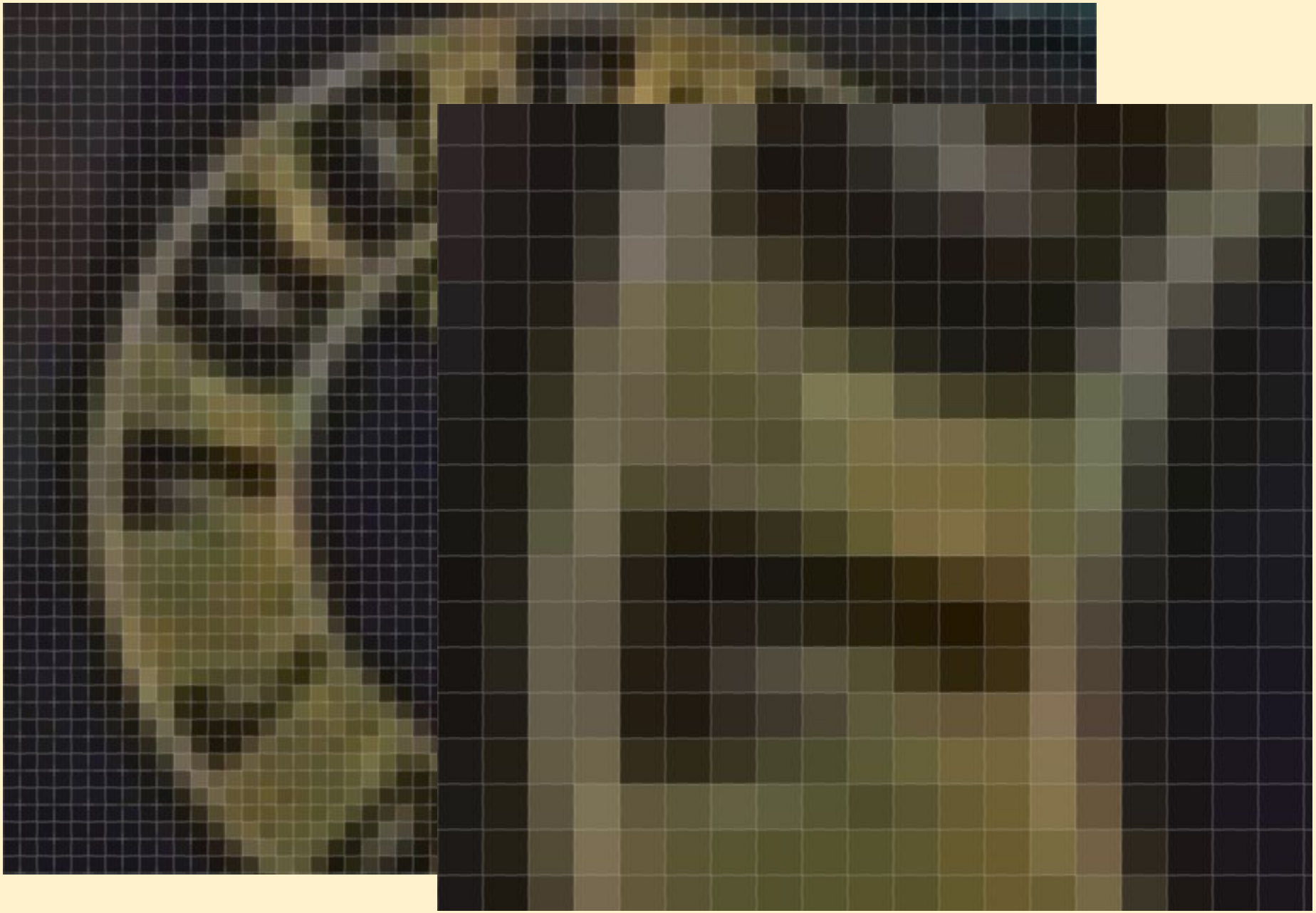
1 = white

Colour bitmaps

Colour images need more bits to represent each pixel.

The more bits you use per pixel the better the colour looks:

- 8 bit greyscale
- 8 bit colour
- 24 bit colour



Colour bitmaps

The **colour depth** is the number of bits used to represent the colour for each pixel in an image

24-bit colour depth means that for each pixel 24 bits are used - providing about **16.7 million colours** - this is standard JPEG colour depth

This is enough colours to fool the human eye into thinking the image is natural

Colour bitmaps

The greater the colour depth the larger the file size.

File size = width x height x colour depth

- Black and white = 1 bit colour depth
- 8 colours = 4 bit colour depth

Image File Sizes

size in bits = width x height x colour depth

size in Bytes = (width x height x colour depth) /8

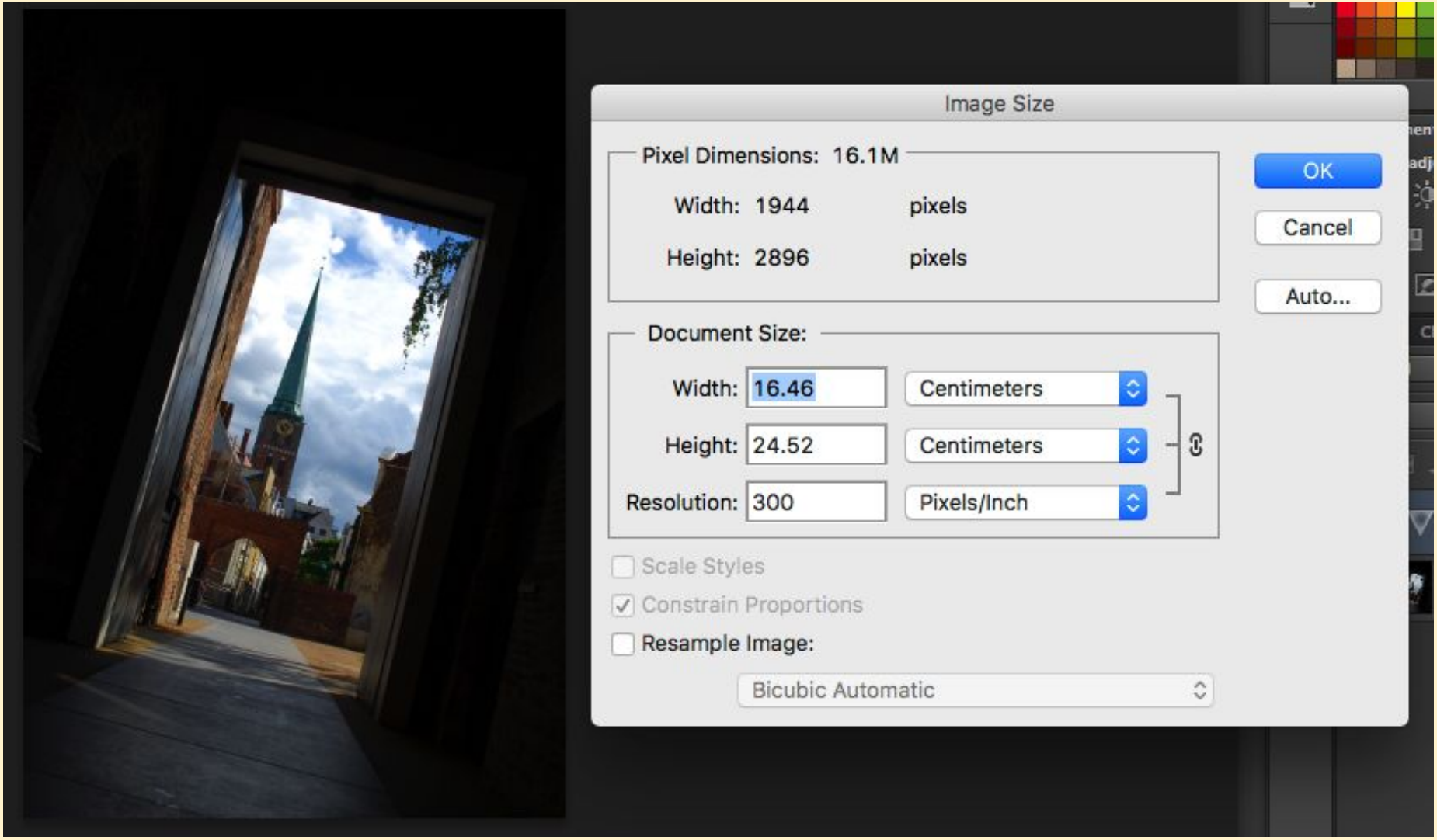


Image size in bits?

Image File Sizes

size in bits = width x height x colour depth

= 1944 x 2896 x 24

=

Image File Sizes

size in bits = width x height x colour depth

= 1944 x 2896 x 24

= 135,115,776 bits

Image File Sizes

size in Bytes = (width x height x colour depth) /8

= 135,115,776 / 8

= 16,889,472 Bytes

How many KiloBytes? MegaBytes? GigaBytes?

Colour bitmaps

24-bit colours are shown using hex codes:

#00FF00

R G B

Why use hexcodes? It’s easier than using binary but clearer (and less prone to error) than using decimal)