# Image Analysis

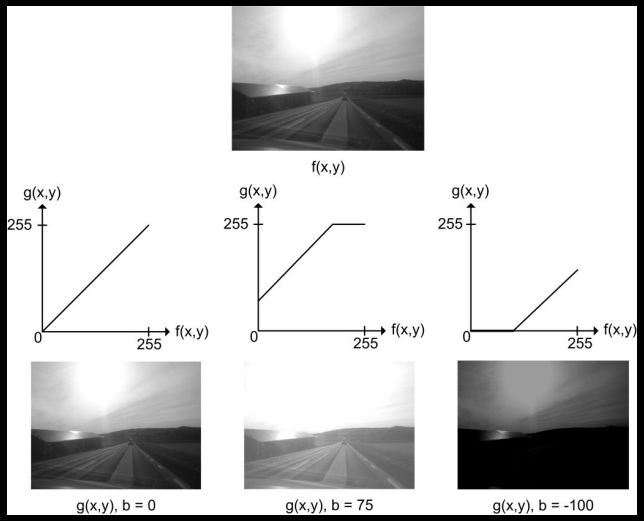
Rasmus R. Paulsen
Tim B. Dyrby

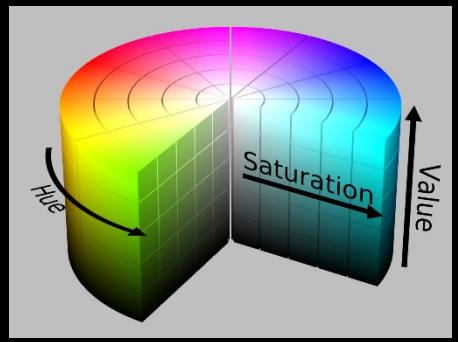
DTU Compute

http://courses.compute.dtu.dk/02502

Plenty of slides adapted from Thomas Moeslunds lectures

# Week 3
# Pixelwise operations and colour images
# PCA on images



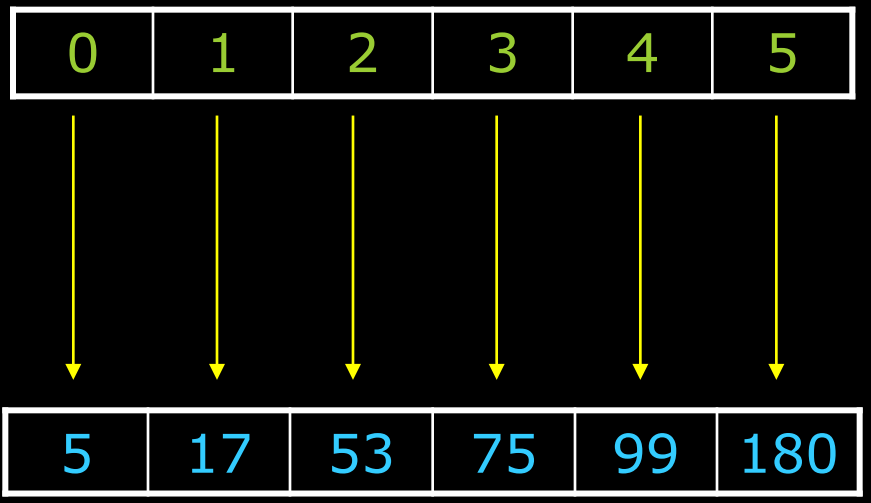
# What can you do after today?

- Compute and apply a linear gray transformation
- Describe and compute the image histogram
- Implement and apply histogram stretching
- Implement and apply gamma transformation
- Implement and apply log and exp mappings
- Describe and use thresholding
- Describe and use automatic thresholding
- Perform conversions between bytes and doubles
- Use addition and subtraction of images
- Explain the benefits of bi-modal histograms
- Identify images where global thresholding can be used for object extraction

# …and you can even more

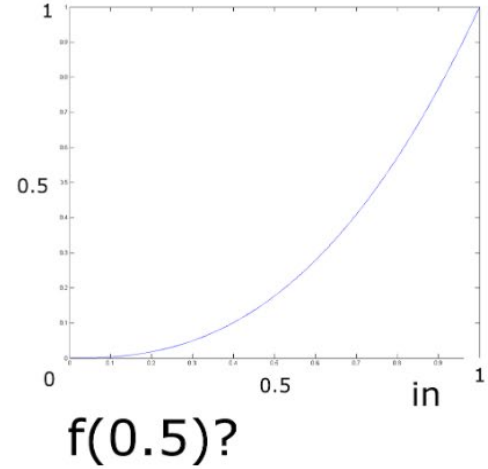
- Describe the basic human visual system including rods and cones
- Describe subtractive colors
- Describe additive colors
- Describe the RGB color space
- Describe the normalised RGB color representation
- Describe the use of the Bayer pattern in digital cameras
- Describe the HSI color space
- Convert from an RGB to a grey level value
- Convert from an RGB value to an HSI value
- Describe the use of different color spaces
- Implement and use color thresholding in RGB space
- Implement and use color thresholding in HSI space

# Gray value mappings



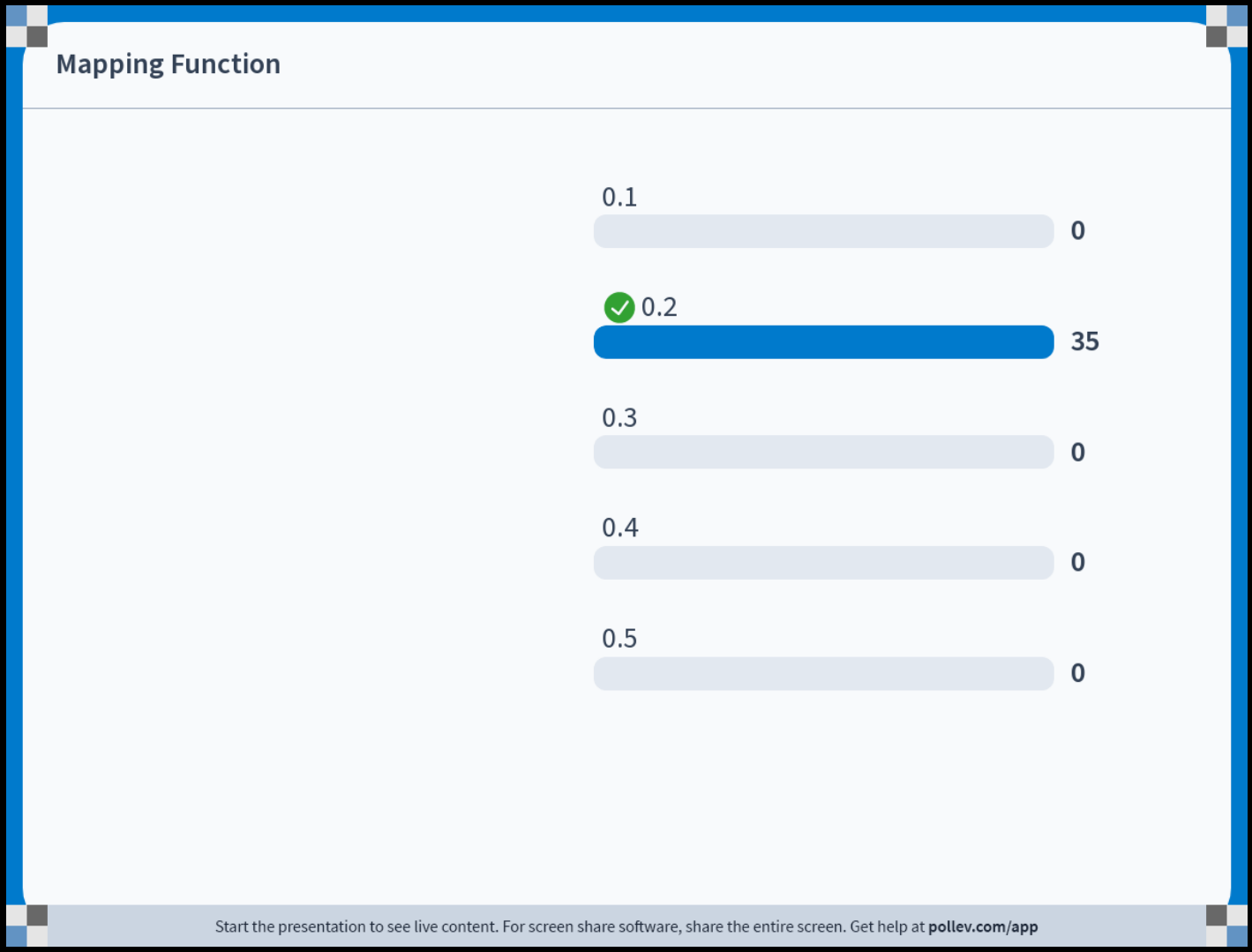

- Mapping
  - To make correspondence between two sets of values
- Look-up-table
  - A table of mappings

## Mapping Function



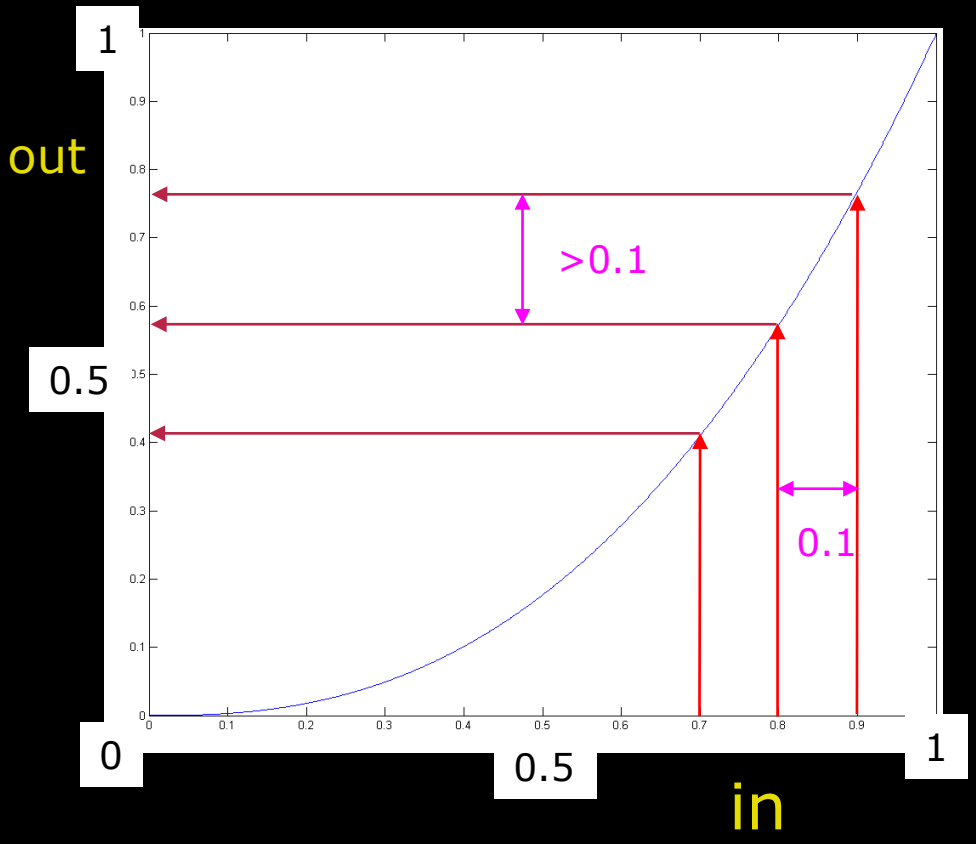

- 0.1
- 0.2
- 0.3
- 0.4
- 0.5

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

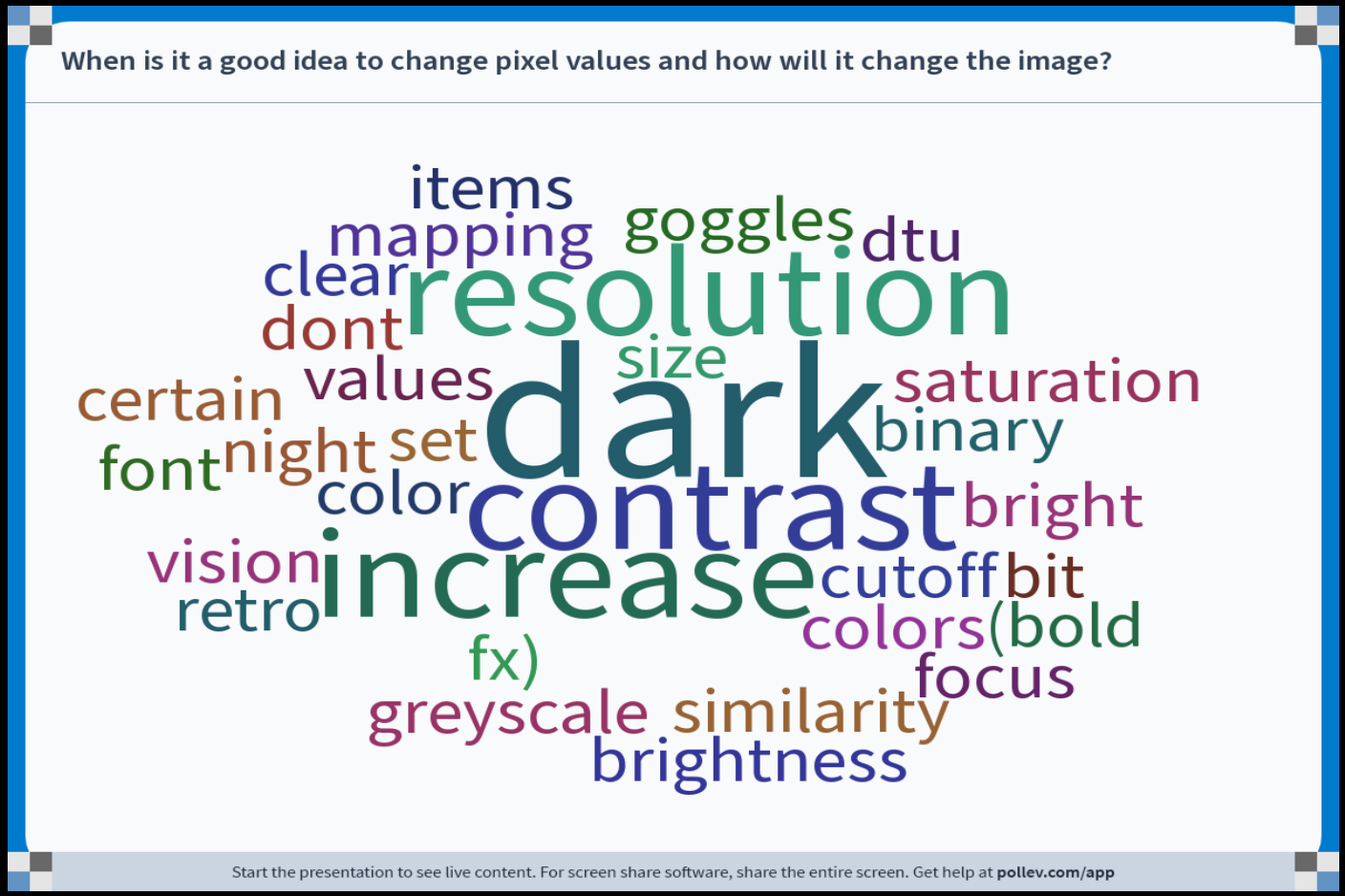

Mapping Function
0.1
0
0.2
35
0.3
0
0.4
0
0.5
0
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Mapping Function
0.1
0
0.2
35
0.3
0
0.4
0
0.5
0
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

# Gray value mappings



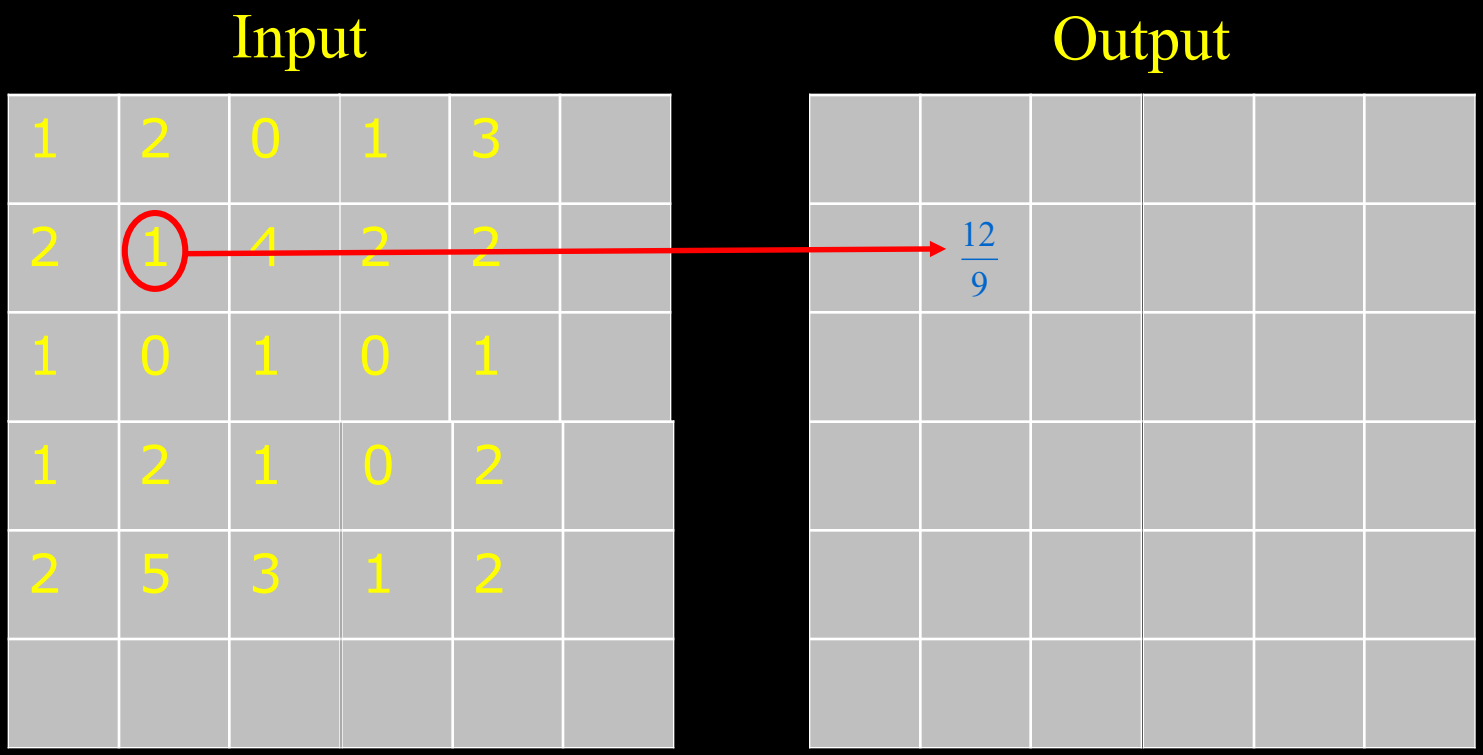

- Mapping
  - To make correspondence between two sets of values
- Mapping function
  - out = f(in)
- What happens with the values?
  - Values with difference 0.1
  - Output values “spread out”


When is it a good idea to change pixel values and how will it change the image?
items
mapping
goggles
dtu
clear
resolution
dont
size
values
dark
saturation
certain
binary
font
night
set
color
contrast
bright
vision
increase
cutoff
bit
retro
colors(bold
fx)
focus
greyscale
similarity
brightness
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

# Why change gray level values

- When could it be good to change the gray level values?
  - Lack of contrast
  - Very dark image
  - Very bright image

# Point processing




- The value of the output pixel is only dependent on the value of one input pixel
- A global operation – changes all pixels

# Point processing

- Grey level enhancement
  - Process one pixel at a time independent of all other pixels
  - For example used to correct Brightness and Contrast
    - Known from the television remote control

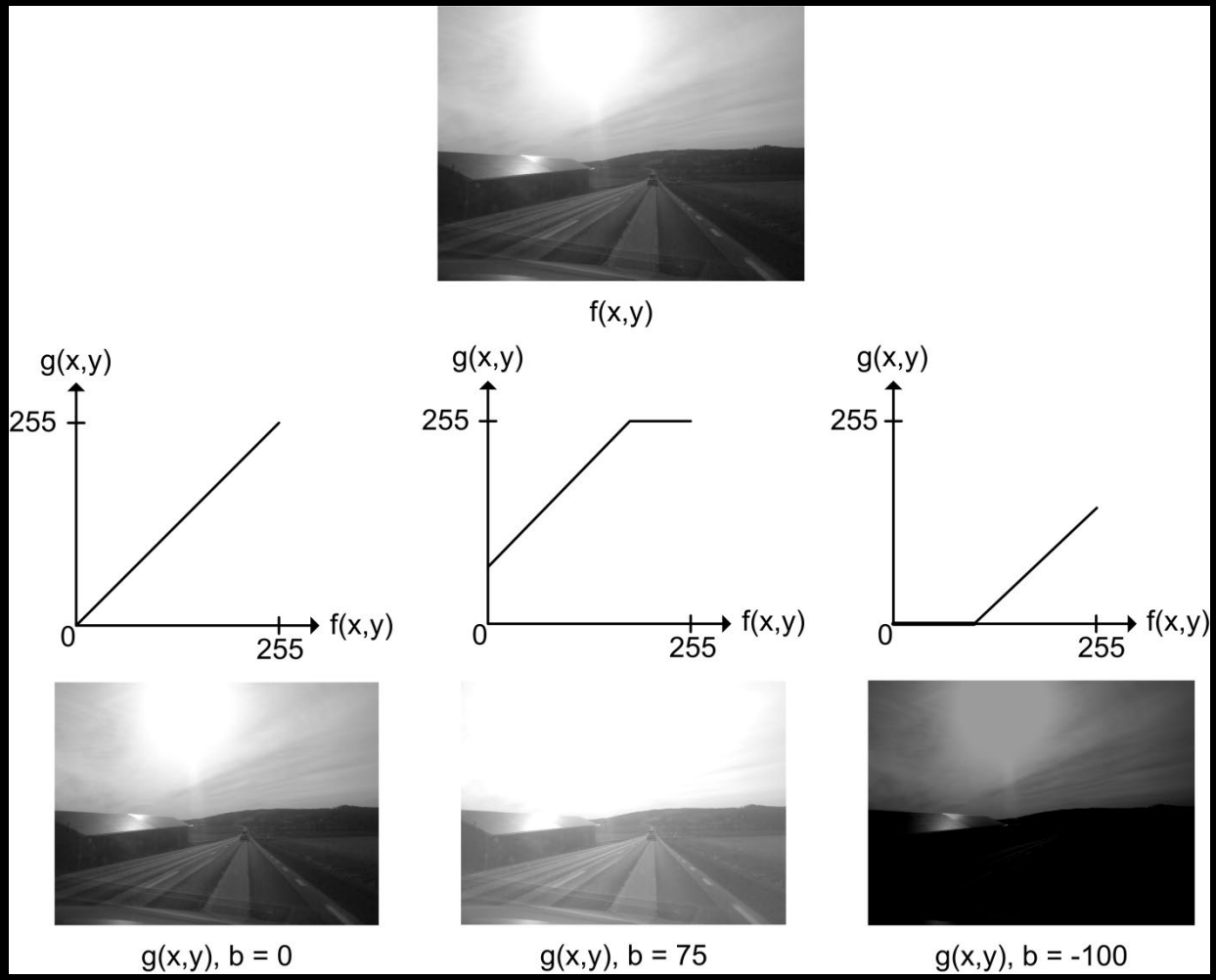
Correct | Too high brightness | Too low brightness | Too high contrast | Too low contrast

# Brightness

- The brightness is the intensity
- Change brightness:
  - To each pixel is added the value *b*
  - f(x, y) is the input image
  - g(x, y) is the (enhanced) output image
- If $b>0$ : brighter image
- If $b<0$ : less bright image

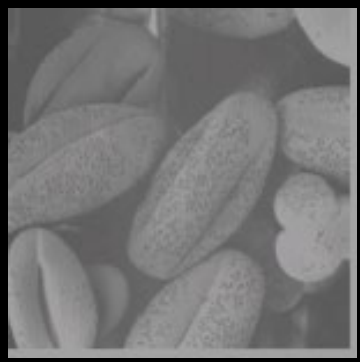
$$g(x,y) = f(x,y) + b$$

# Brightness


f(x,y)
g(x,y)
255
0
255
f(x,y)
g(x,y)
255
0
255
f(x,y)
g(x,y)
255
0
255
f(x,y)
g(x,y), b = 0
g(x,y), b = 75
g(x,y), b = -100

# Contrast

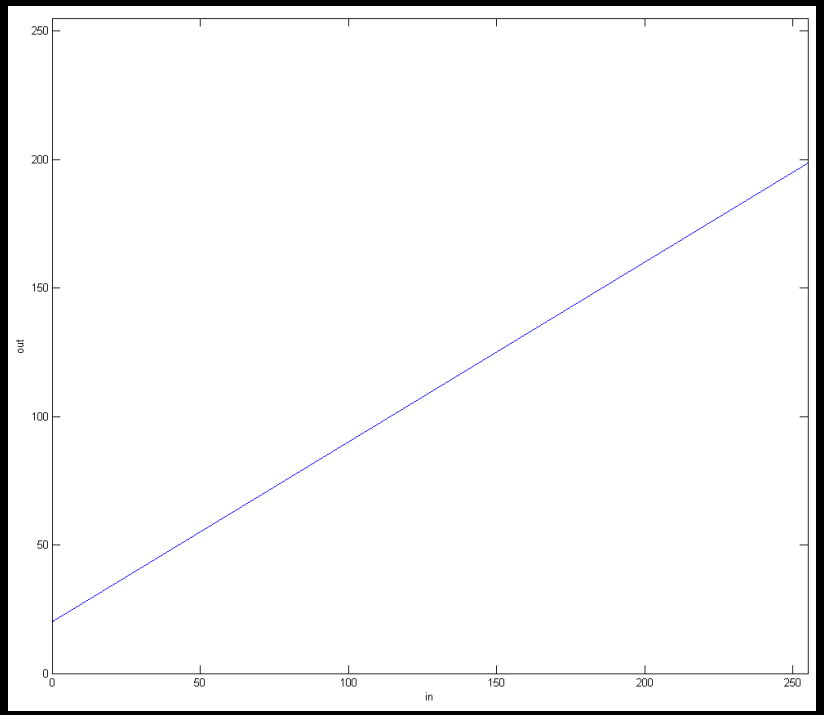
- The contrast describes the level of details we can see
- Change contrast
- Each pixel is multiplied by *a*
  - f(x, y) is the input image
  - g(x, y) is the (enhanced) output image
- If *a*>1 => more contrast
- If *a*<1 => less contrast



$$g(x,y) = a * f(x,y)$$

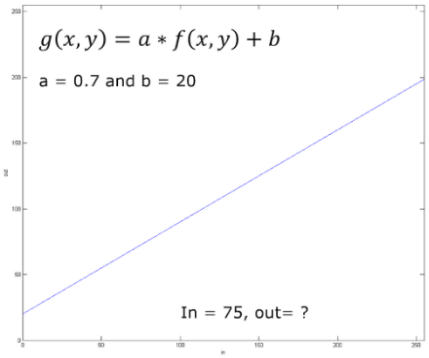
# Combining brightness and contrast

- A straight line
- Called a *linear transformation*
- Here a = 0.7 and b = 20



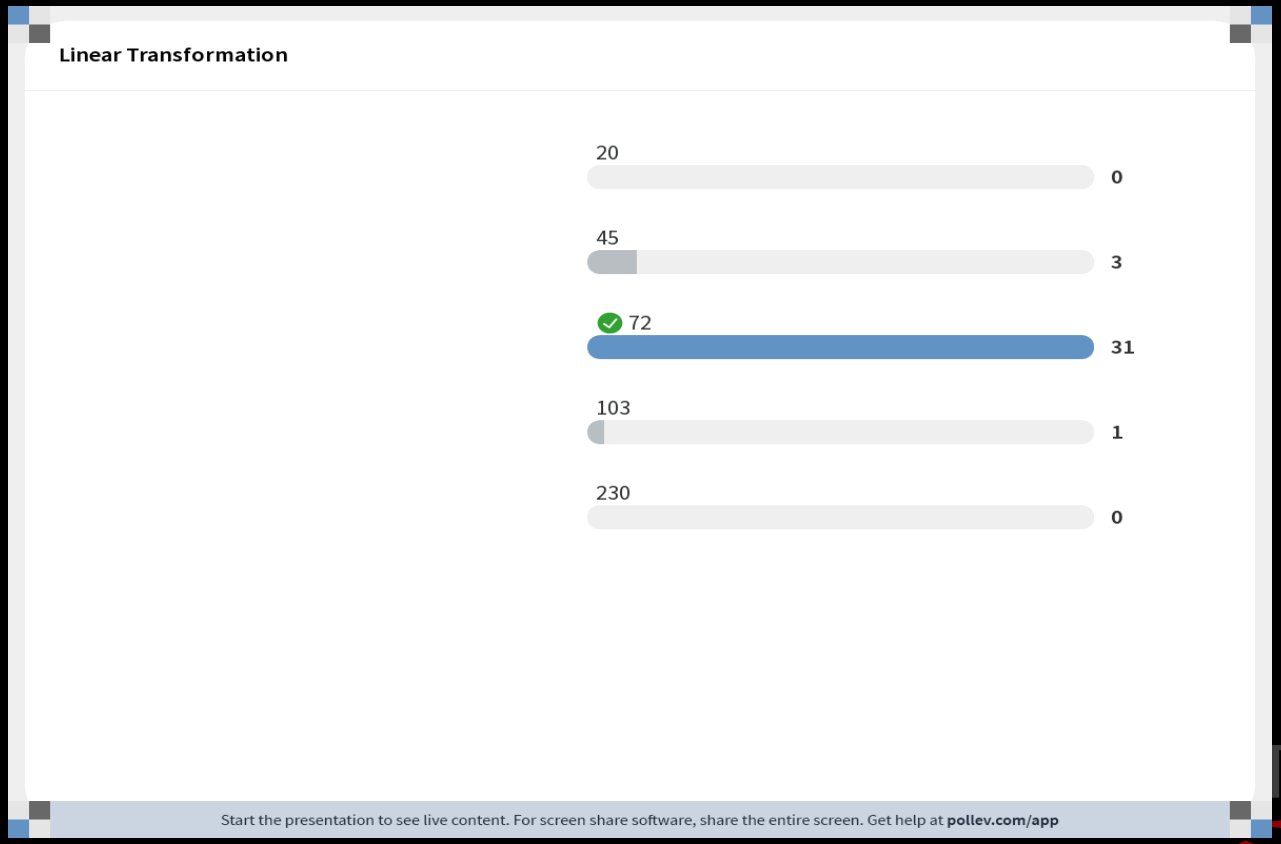

$$g(x,y) = a * f(x,y) + b$$

**Linear Transformation**

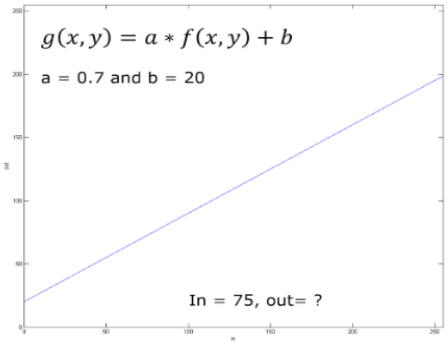



- 20
- 45
- 72
- 103
- 230

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**


Linear Transformation
20
0
45
3
72
31
103
1
230
0
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

**Linear Transformation**

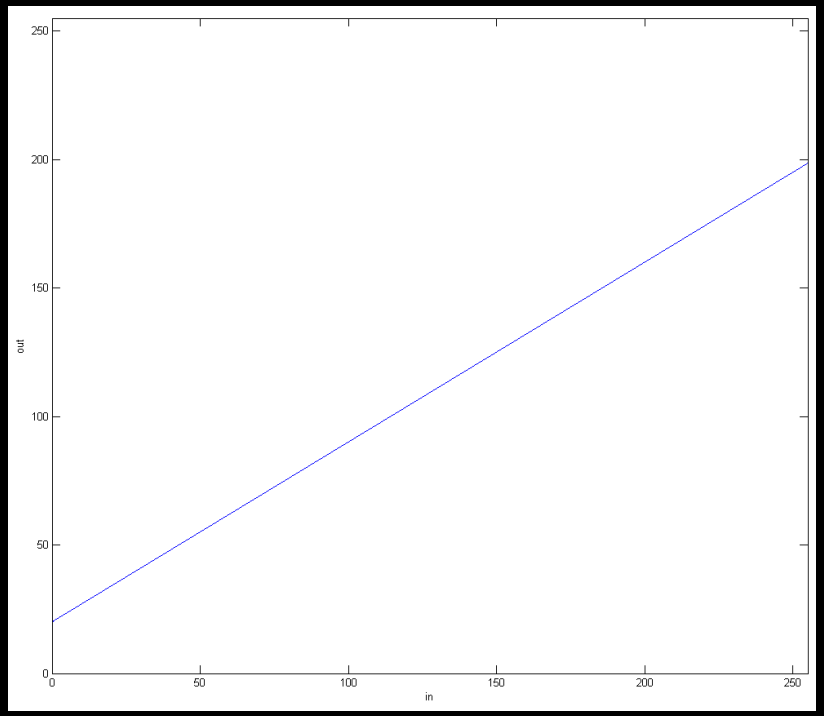



| | |
|---|---|
| 20 | 0 |
| 45 | 3 |
| ✓ 72 | 31 |
| 103 | 1 |
| 230 | 0 |

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

# Combining brightness and contrast

- A straight line
- Called a *linear transformation*
- Here a = 0.7 and b = 20
- What will the visual result be on the output image?
  - More bright (b > 0)
  - Less contrast (a < 1)



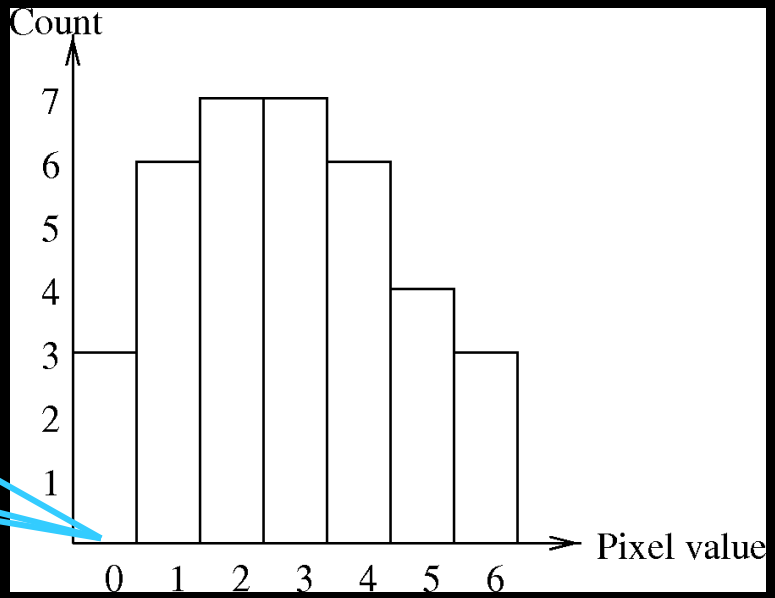

$$g(x,y) = a * f(x,y) + b$$

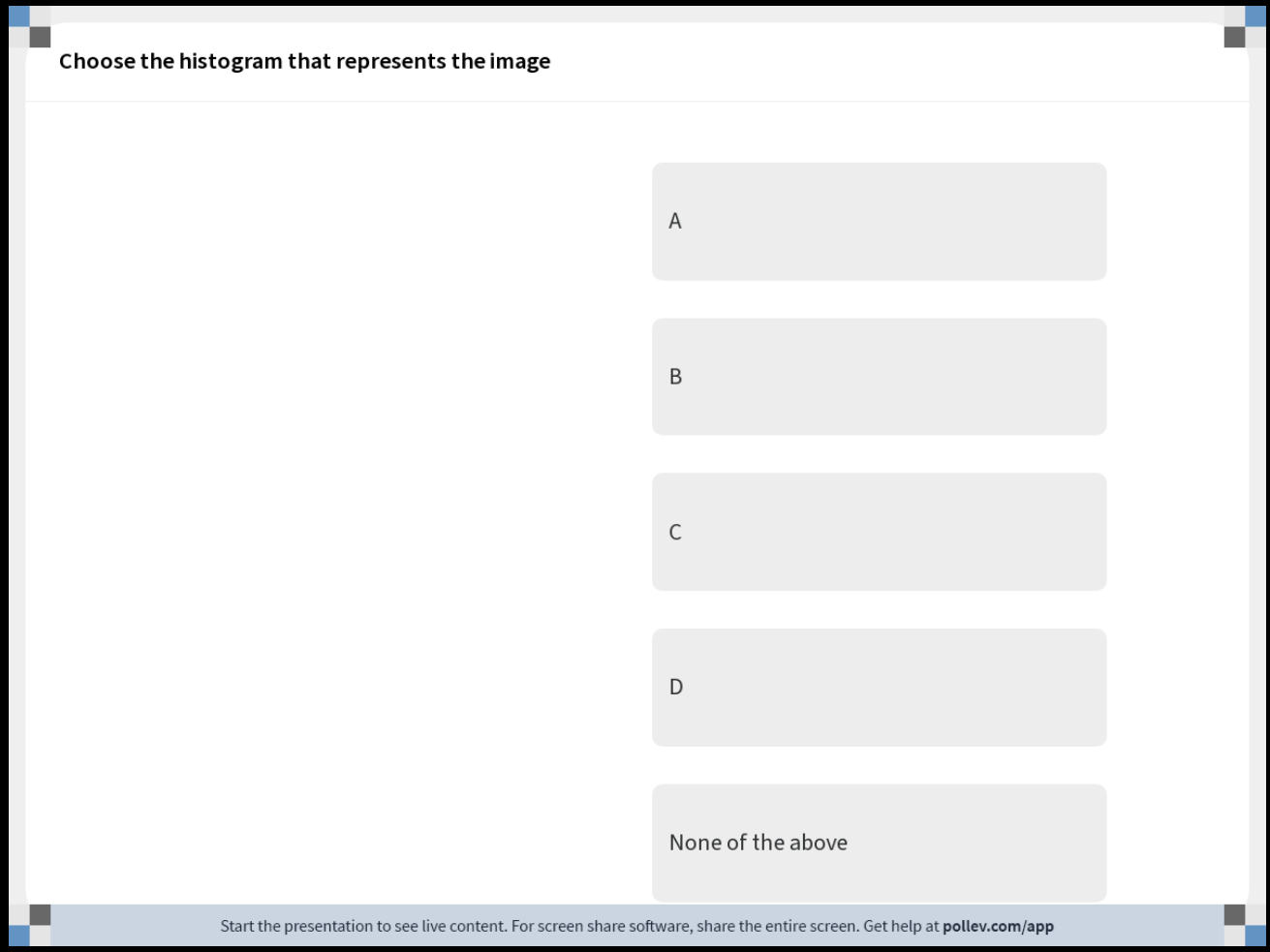
# Histogram Reminder

- A histogram normally contains the same number of "bins" as the possible pixel values
- A bin stores the number of pixel with that value

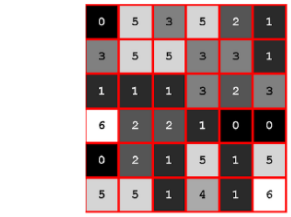
| | | | | | |
|---|---|---|---|---|---|
| 0 | 2 | 6 | 6 | 3 | 3 |
| 1 | 4 | 3 | 4 | 4 | 4 |
| 3 | 2 | 5 | 1 | 5 | 2 |
| 1 | 4 | 2 | 1 | 3 | 1 |
| 2 | 5 | 3 | 0 | 2 | 0 |
| 4 | 2 | 5 | 6 | 3 | 1 |

3

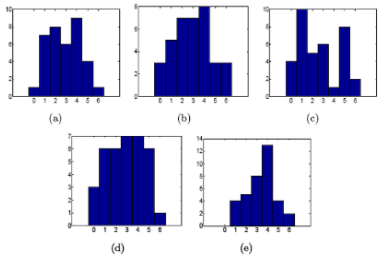



Choose the histogram that represents the image
A
B
C
D
None of the above
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

**Choose the histogram that represents the image**




Figur 6: Grayscale billede.




A 0

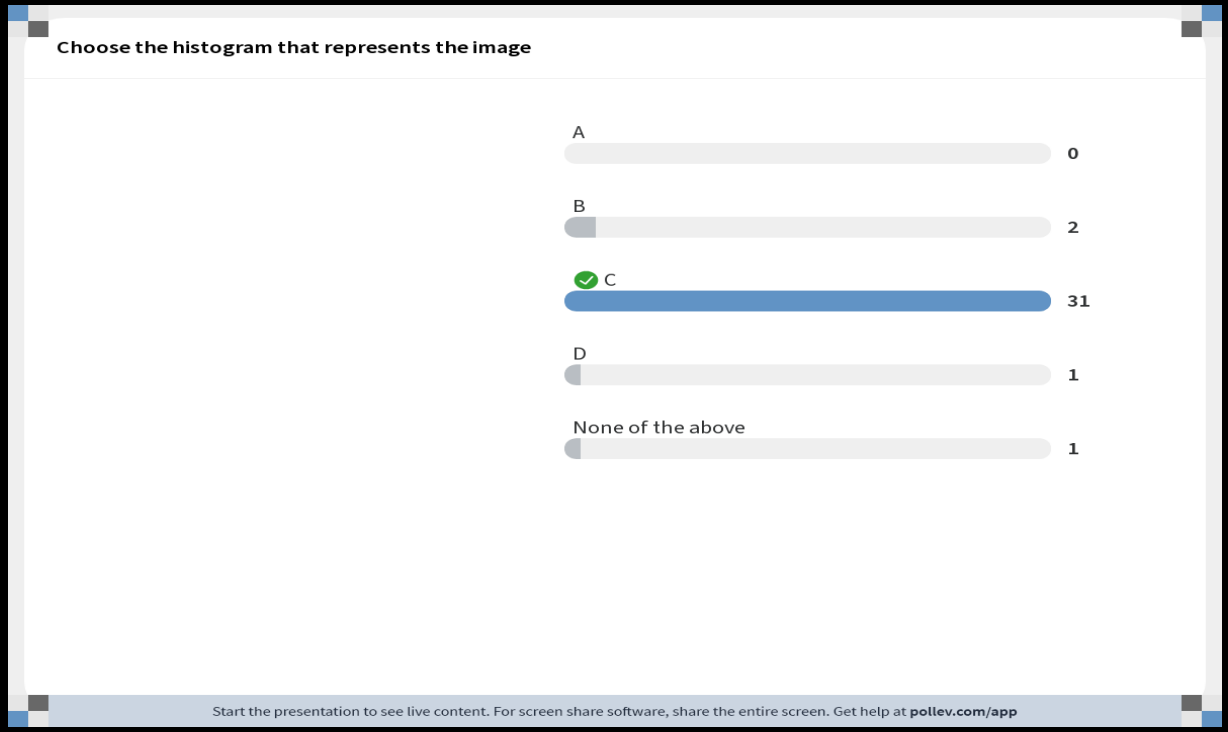
B 2

C 31

D 1

None of the above 1

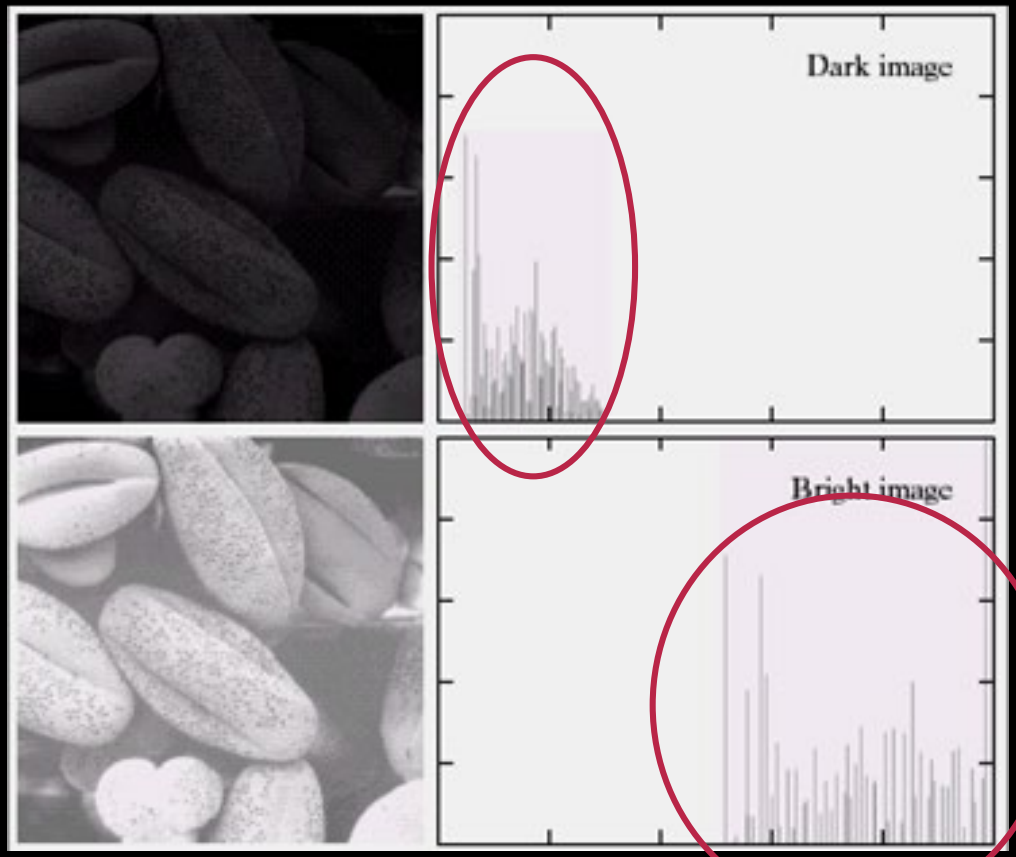
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**


Choose the histogram that represents the image
A
0
B
2
C
31
D
1
None of the above
1
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

# Back to the histogram

- The shape of the histogram tells us a lot!

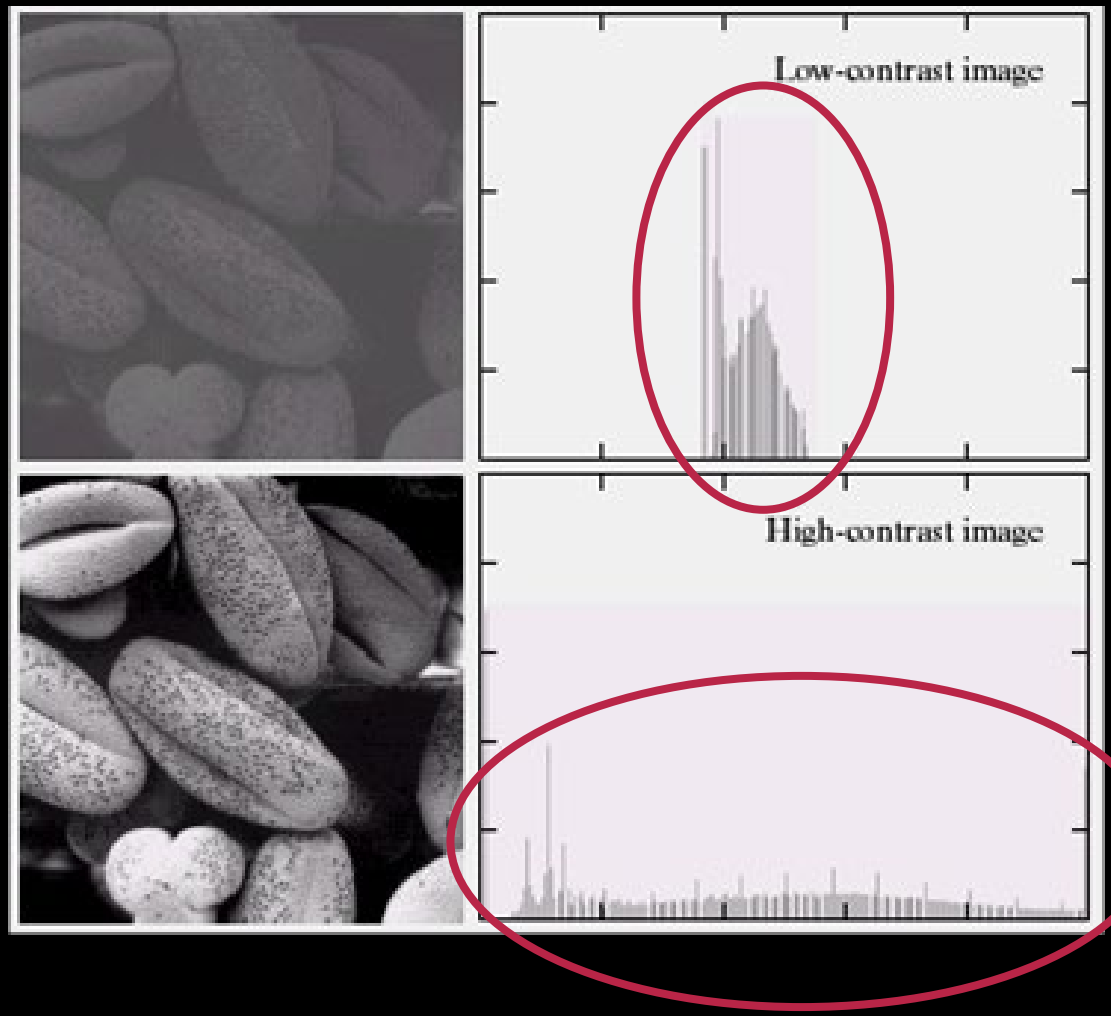
# Histogram inspection




Dark image

Bright image

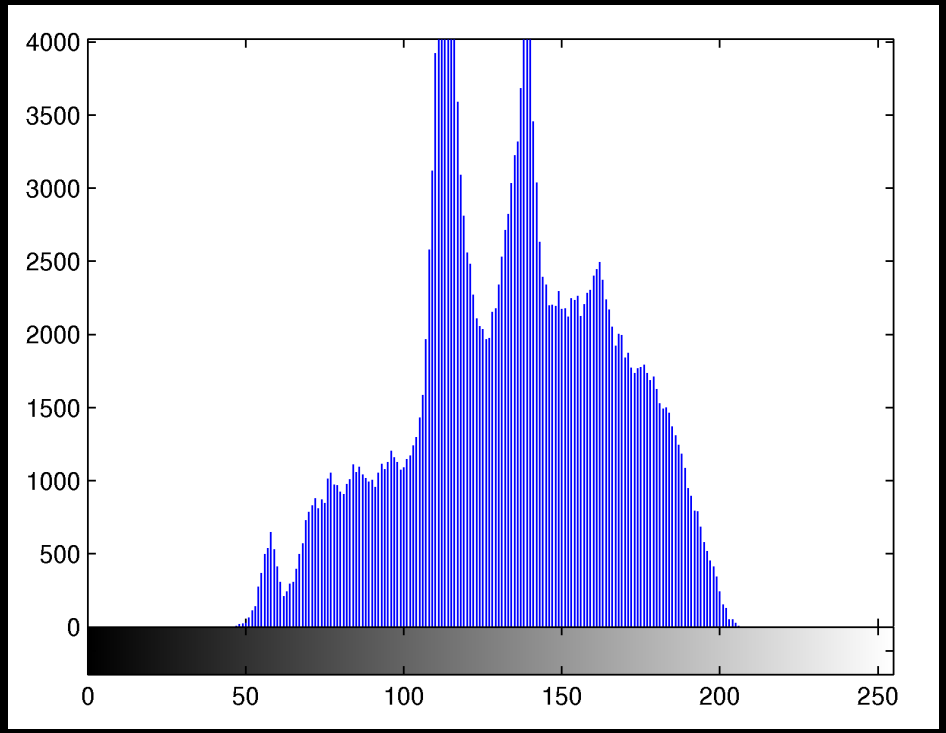
# Histogram inspection




Low contrast

High contrast

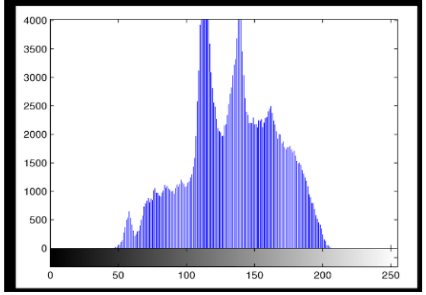
# Histogram stretching




- How do we optimise the image using the histogram?
  - Minimum and maximum values?
  - Stretch it so new minimum = 0 and new maximum = 255

**Histogram stretching**

- We want
  - Min = 0
  - Max = 255
- We have
  - Min = 32
  - Max = 208



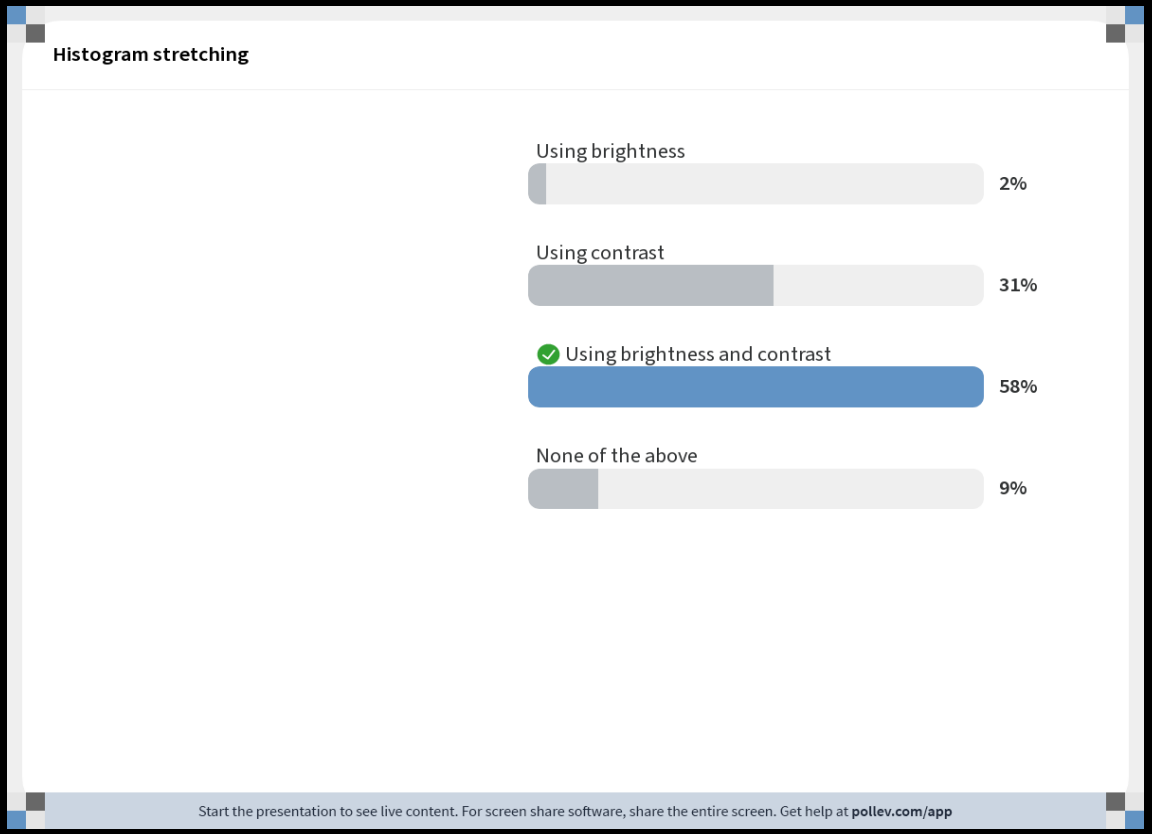

Using brightness

Using contrast

Using brightness and contrast

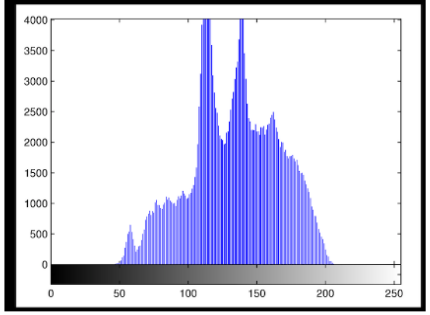
None of the above

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**


Histogram stretching
Using brightness
2%
Using contrast
31%
Using brightness and contrast
58%
None of the above
9%
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

**Histogram stretching**

- We want
  - Min = 0
  - Max = 255
- We have
  - Min = 32
  - Max = 208




Using brightness 2%

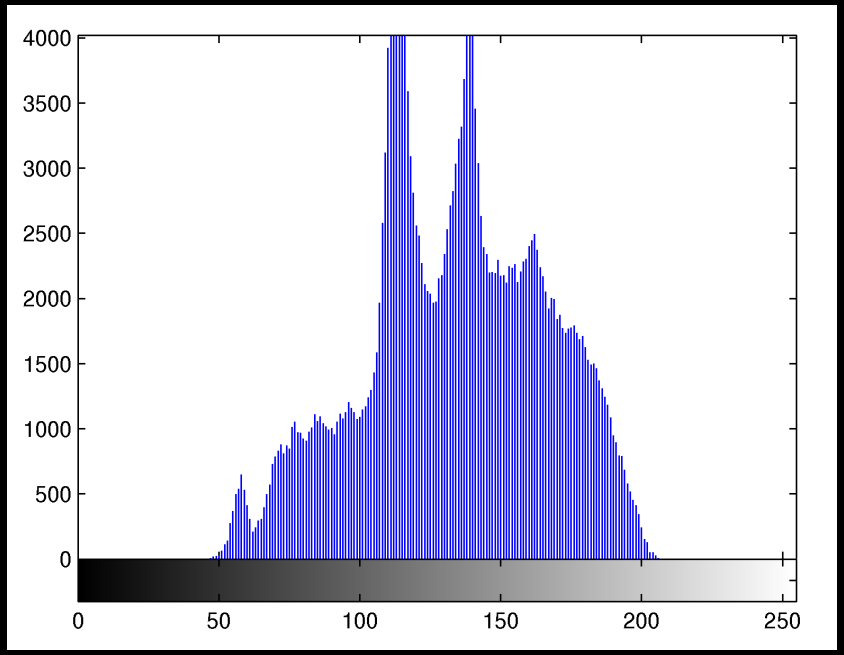
Using contrast 31%

Using brightness and contrast 58%

None of the above 9%

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

# Histogram stretching




- We want
  - Min = 0
  - Max = 255
- We have
  - Min = 32
  - Max = 208

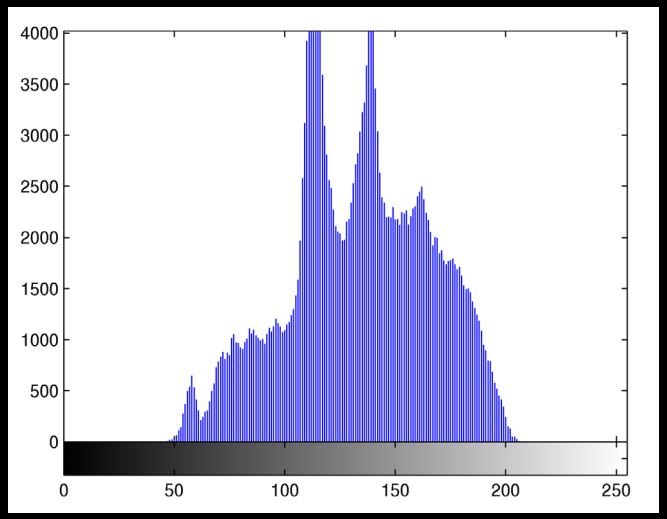
$$g(x,y) = \frac{v_{max,d} - v_{min,d}}{v_{max} - v_{min}} (f(x,y) - v_{min}) + v_{min,d}$$

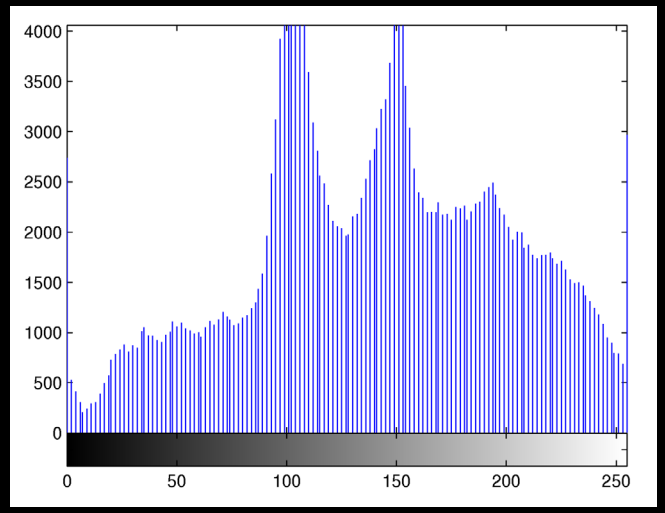
# Histogram stretching formula

$$g(x,y) = \frac{v_{max,d} - v_{min,d}}{v_{max} - v_{min}} (f(x,y) - v_{min}) + v_{min,d}$$

- Desired min value $v_{min,d} = 0$
- Desired max value $v_{max,d} = 255$
- Current min value $v_{min} = 32$
- Current max value $v_{max} = 208$

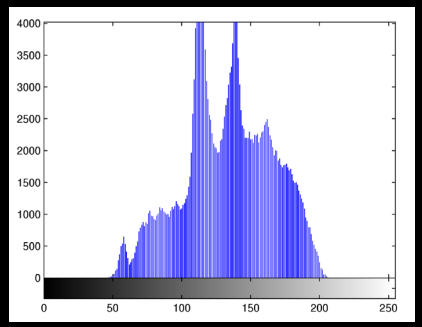
# Histogram stretching



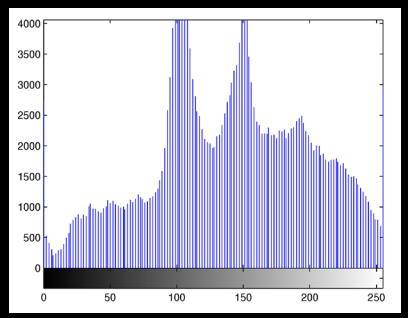

$$g(x,y) = \frac{255}{176}(f(x,y) - 32)$$

# Effect of histogram stretching




32

208

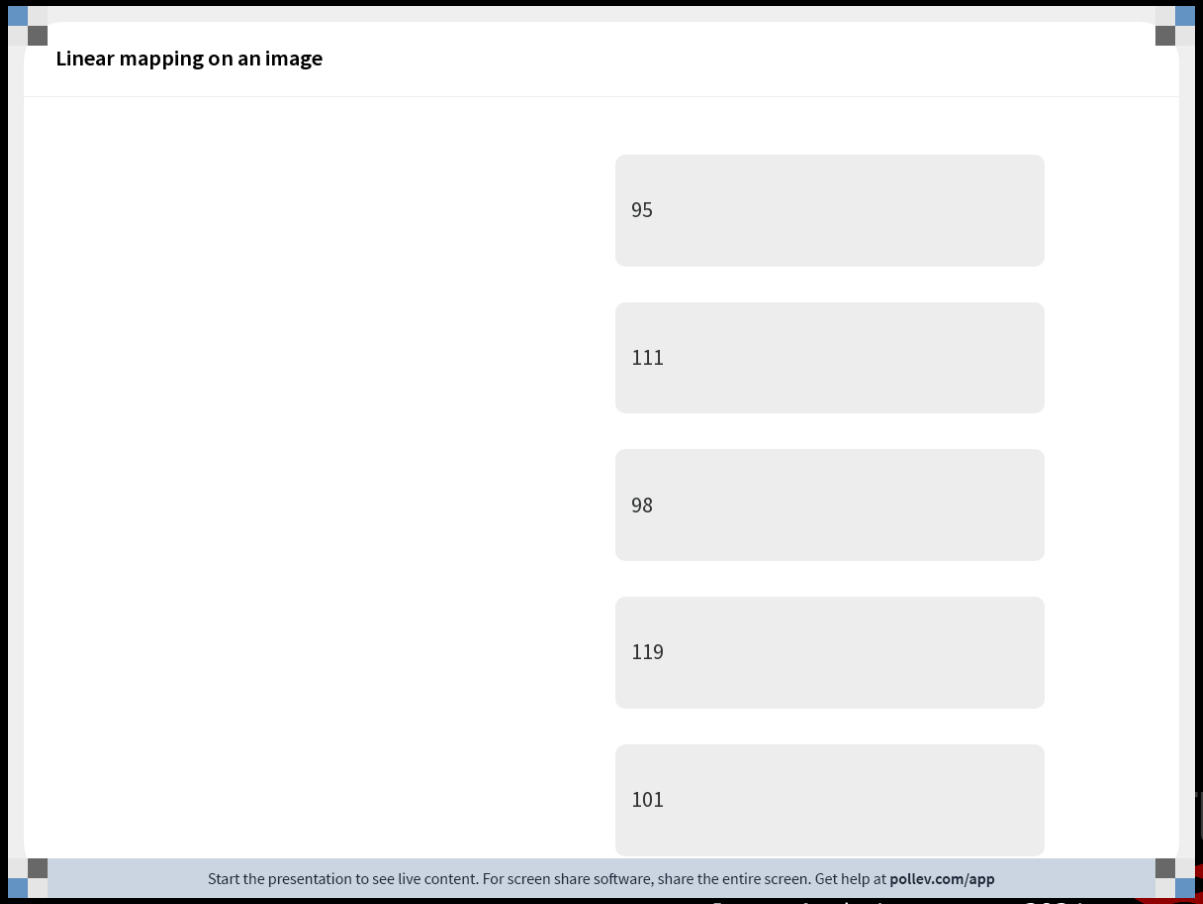


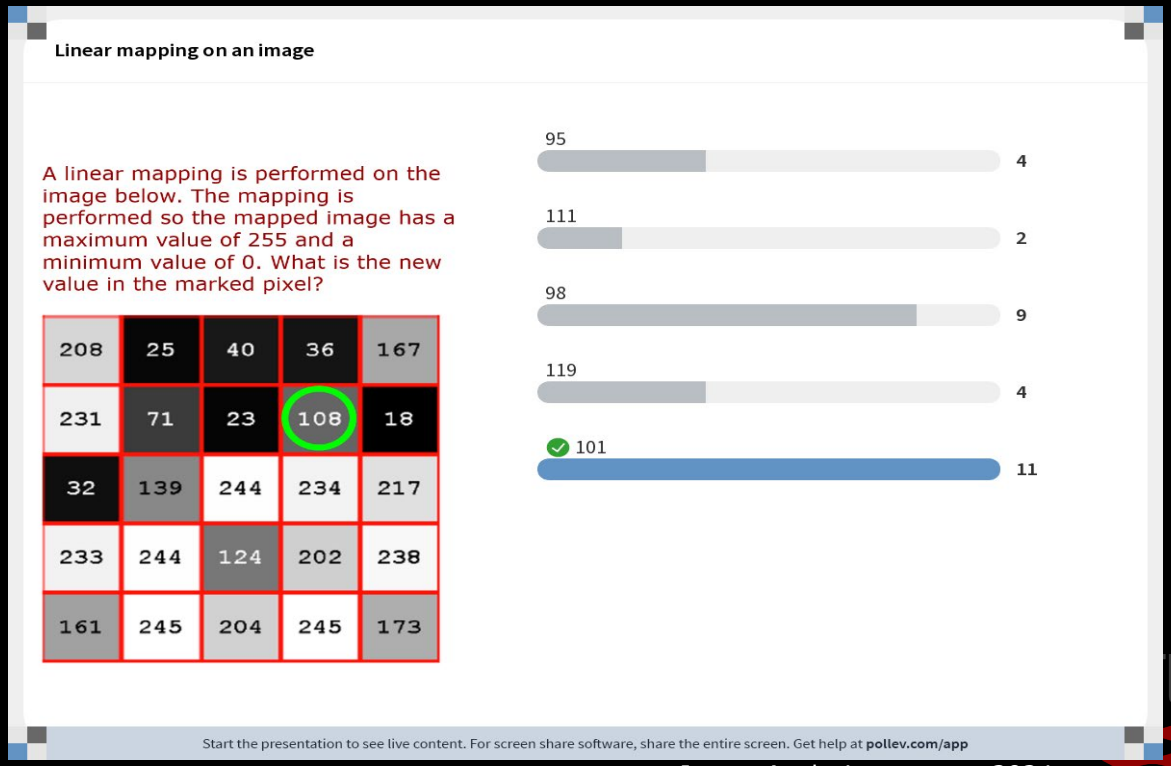

0

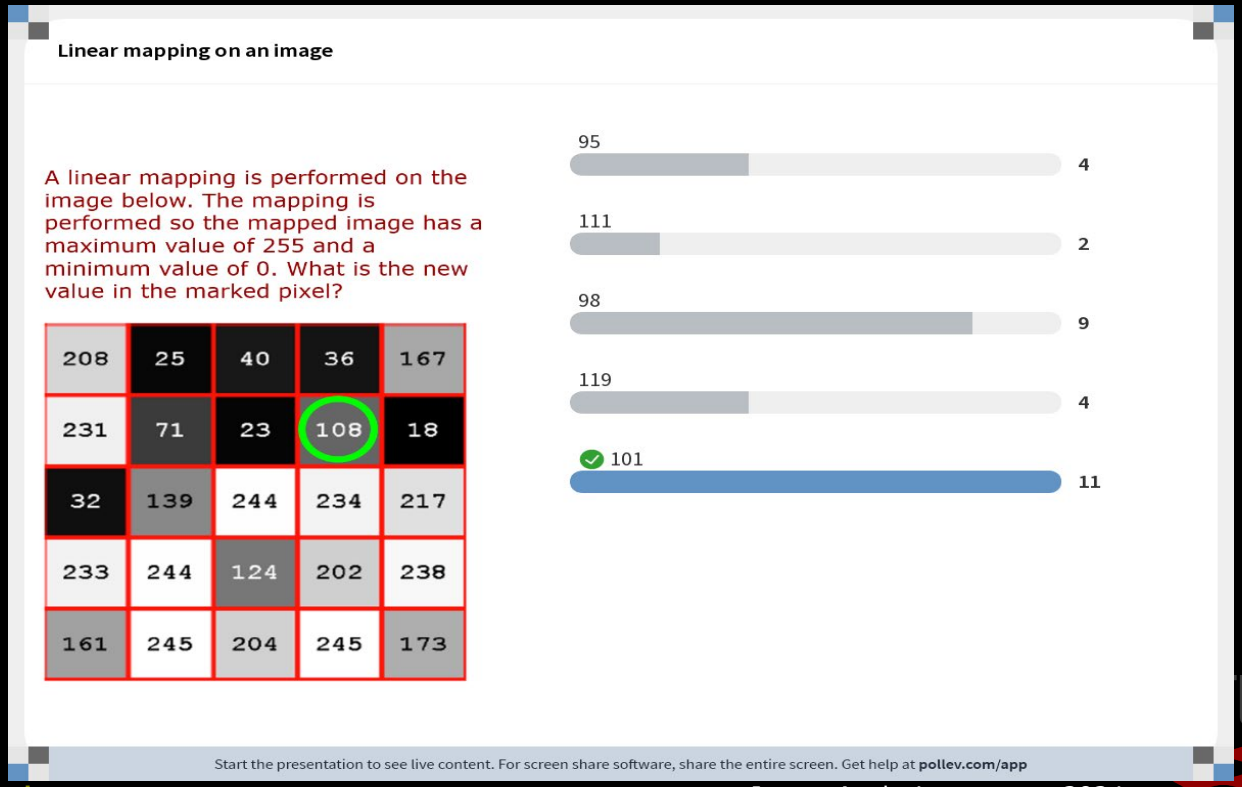
255

# Histogram stretching – weaknesses

- A single pixel value of 0 or 255 ruins it
- Sometimes you want
  - To stretch only the high pixel values
  - While "compressing" the low pixel values
  - Non-linear mapping


Linear mapping on an image
95
111
98
119
101
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

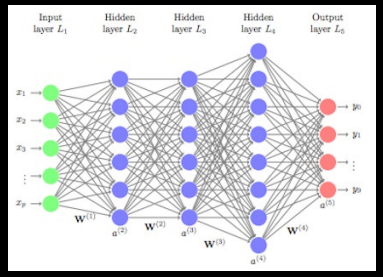

Linear mapping on an image
A linear mapping is performed on the image below. The mapping is performed so the mapped image has a maximum value of 255 and a minimum value of 0. What is the new value in the marked pixel?
208 25 40 36 167
231 71 23 108 18
32 139 244 234 217
233 244 124 202 238
161 245 204 245 173
95 4
111 2
98 9
119 4
101 11
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Linear mapping on an image
A linear mapping is performed on the image below. The mapping is performed so the mapped image has a maximum value of 255 and a minimum value of 0. What is the new value in the marked pixel?
208 25 40 36 167
231 71 23 108 18
32 139 244 234 217
233 244 124 202 238
161 245 204 245 173
95 4
111 2
98 9
119 4
101 11
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

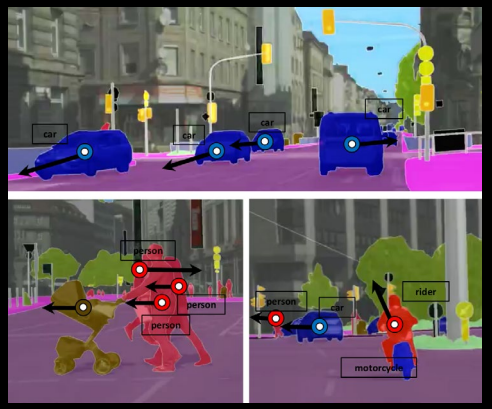
# Deep learning and color/gray scale transformations

- Deep learning needs training data
  - Input image
  - Ground truth labels or classes

- When you lack data you can *augment* your data
  - Create artificial versions
  - Adding variation
  - Changing gray / color levels in the image
  - Point wise operations




http://uc-r.github.io/feedforward_DNN

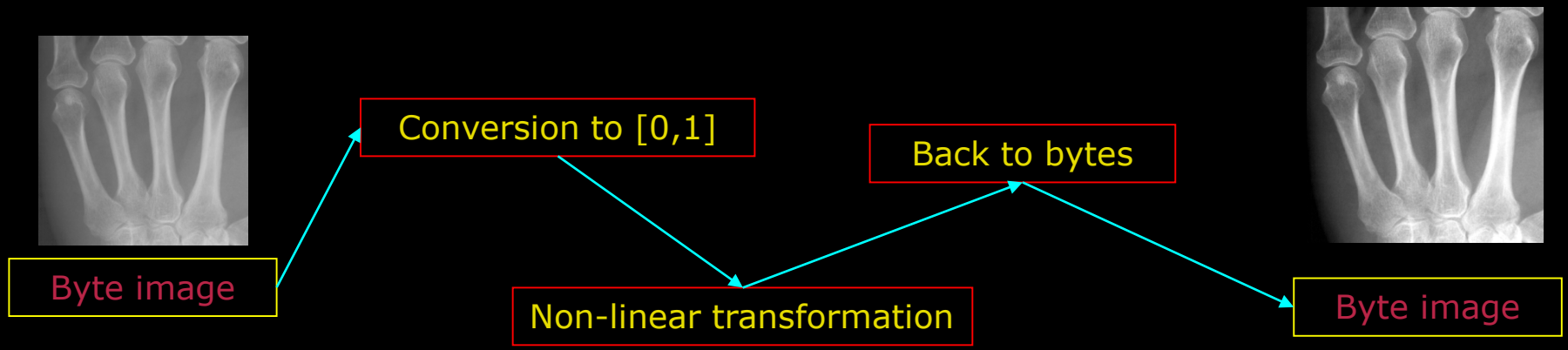



Luc, Pauline, et al. "Predicting deeper into the future of semantic segmentation." IEEE International Conference on Computer Vision (ICCV). Vol. 1. 2017.


https://www.quora.com/What-does-the-term-semantic-segmentation-mean-in-the-context-of-Deep-Learning

# Other mappings

- Non-linear mappings
- Not always nice to work with byte images
  - Better to work with image with values in [0,1]



# Working with bytes and doubles

- A byte contains integer values [0,255]
  - A byte can not store 127.4232
- A value of type *double* can contain “all numbers”
- Why not use doubles always?
  - One double = 8 bytes in the memory
  - Images become very large!
  - Many things can be done with bytes

# Map pixels to [0,1]

- Simple conversion to [0,1]

$$g(x,y) = \frac{1}{255} f(x,y)$$

# Pixels back to bytes

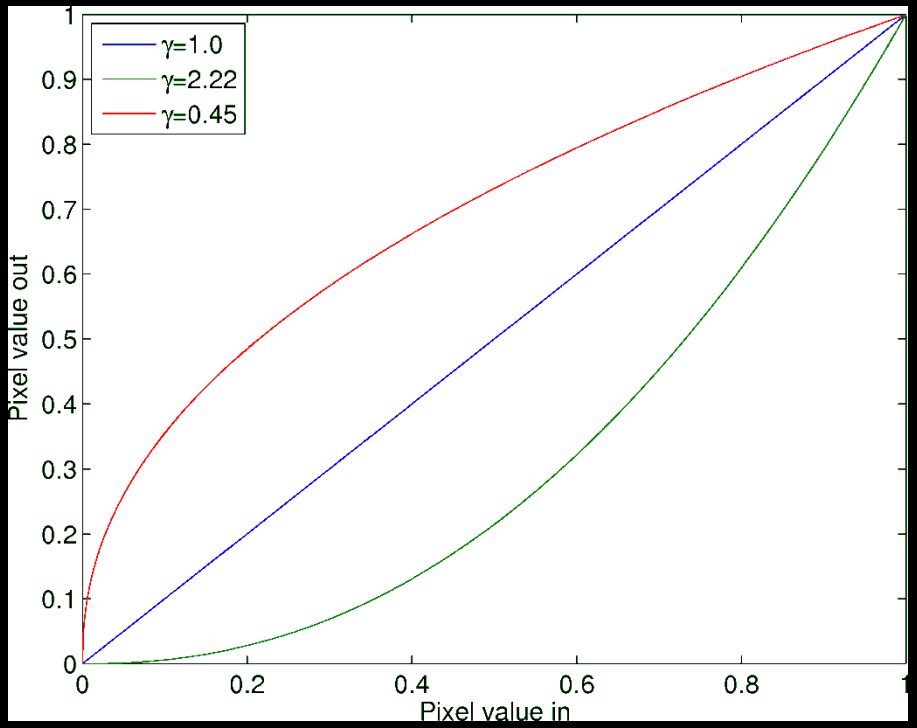
- Input pixels are [0,1]
- We want them to be [0,255]
- Simple linear transformation:

$$g(x,y) = 255 * f(x,y)$$

# Gamma mapping

- Gamma mapping is used in televisions and flat panels
- Can increase the contrast (dynamics) in more selected part of the histogram
- Many games have a possibility for a gamma correction

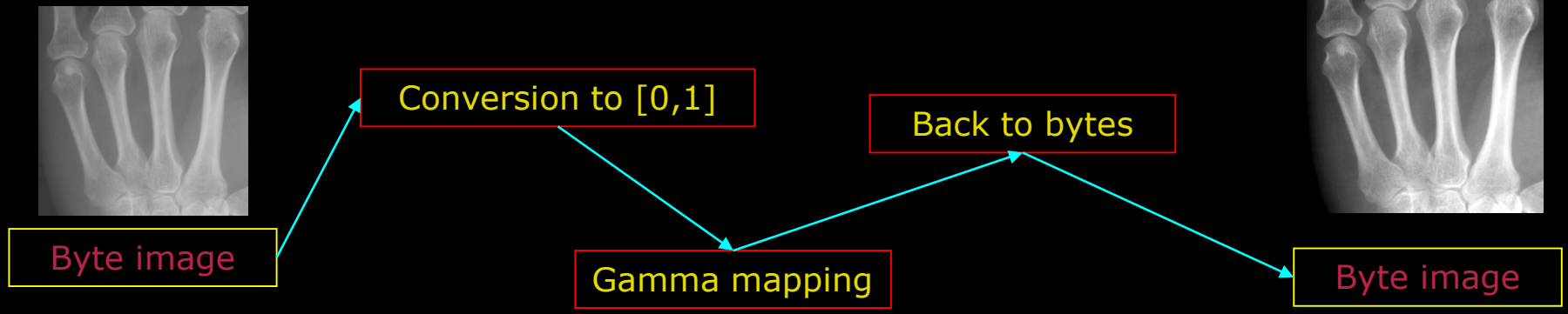
# Gamma curves




- Named after the Greek letter gamma
- What happens to the dark areas
  - With 0.45?
  - With 2.22?

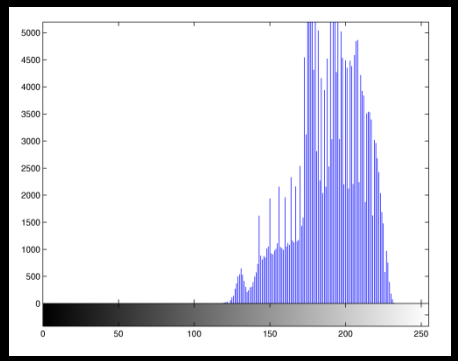
$$g(x,y) = f(x,y)^{\gamma}$$

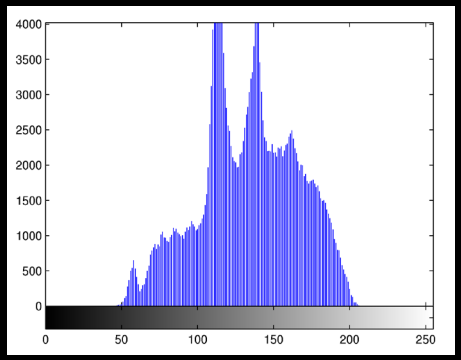
# Perform the gamma mapping



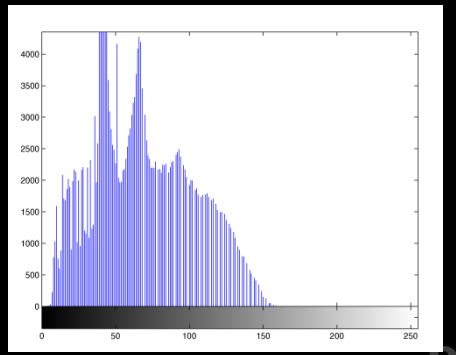
# Results of gamma mapping

0.45

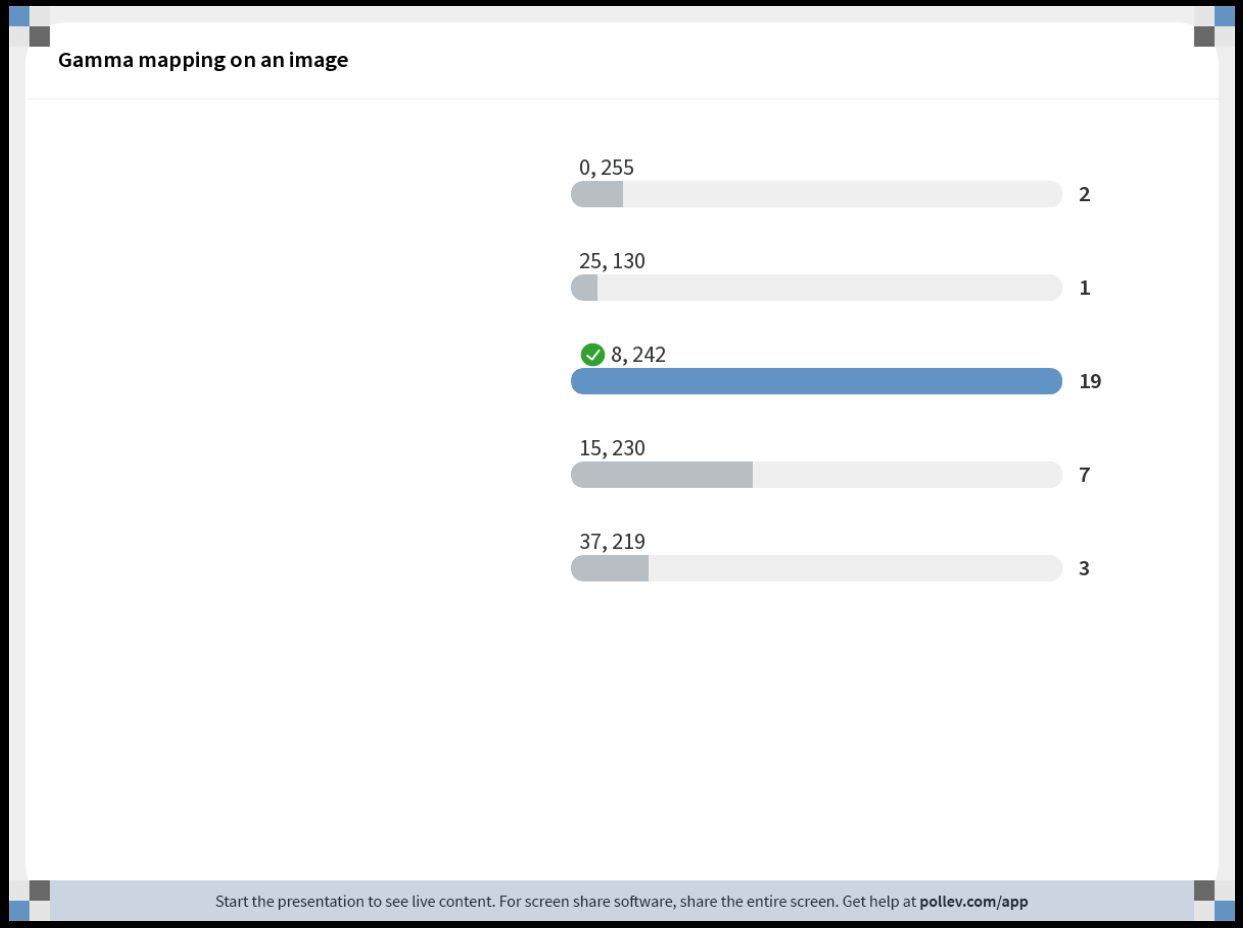



2.22



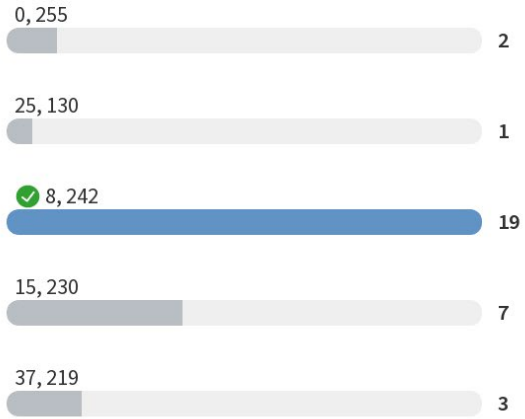

Gamma mapping on an image
0, 255
25, 130
8, 242
15, 230
37, 219
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Gamma mapping on an image
0, 255
2
25, 130
1
8, 242
19
15, 230
7
37, 219
3
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

**Gamma mapping on an image**

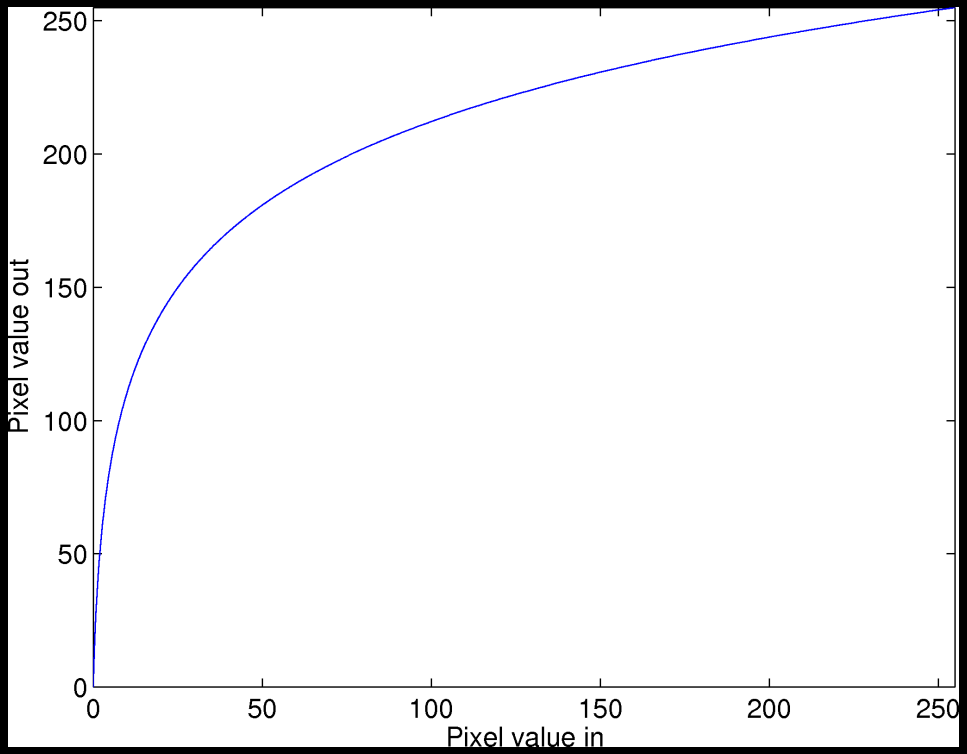
A gamma mapping is performed on the image below with $\gamma = 1.3$. What is the minimum and maximum value in the mapped image?

| | | | | |
|---|---|---|---|---|
| 208 | 25 | 40 | 36 | 167 |
| 231 | 71 | 23 | 108 | 18 |
| 32 | 139 | 244 | 234 | 217 |
| 233 | 244 | 124 | 202 | 238 |
| 161 | 245 | 204 | 245 | 173 |




Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

# Logarithmic mapping



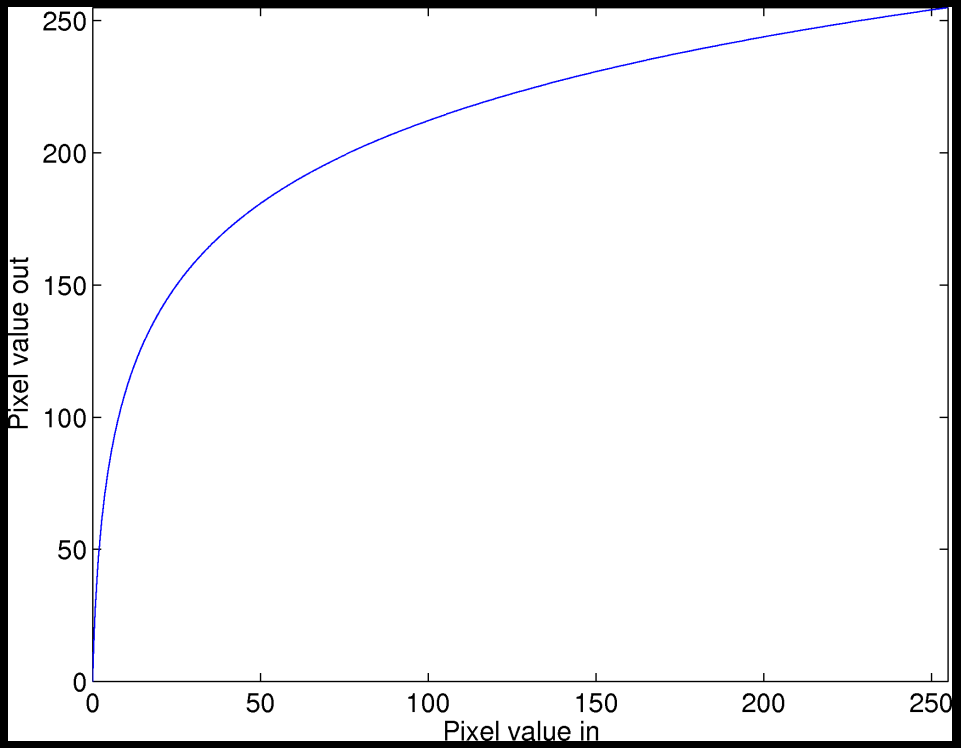

Maps from [0,255] to [0,255]

Why?

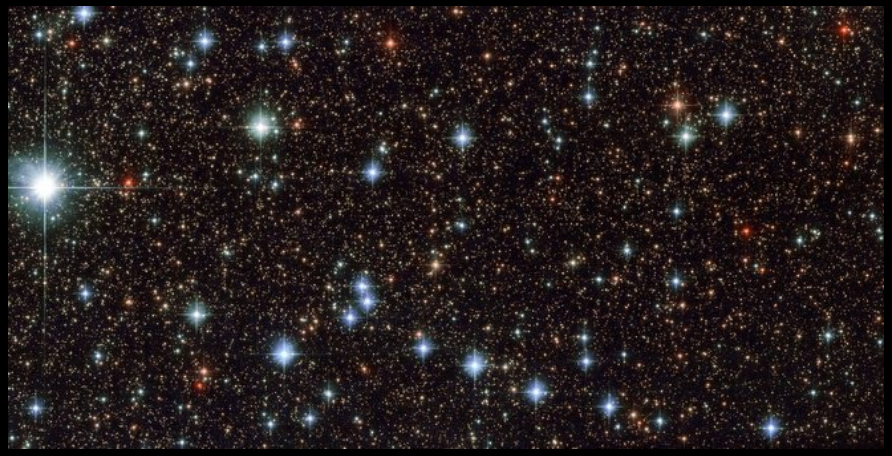
$$g(x,y) = c \log(1 + \mathrm{f}(\mathrm{x},\mathrm{y}))$$

$$c = \frac{255}{\log(1 + v_{max})}$$

# Logarithmic mapping – when?




- For images with very bright spots
- Low intensity pixel values are enhanced

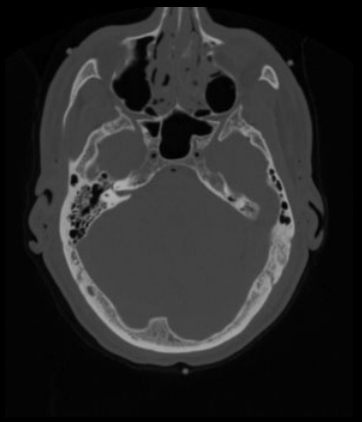


# What do we get out of pixel mappings

- Spreading out or compressing pixel values
  - Better for humans to see
  - New information – no!

# Now for something different

- Until now image processing
  - Input image transformed to output image
- Now for something more like image analysis
- Segmentation
  - Segment the image into regions
    - Background and objects for example

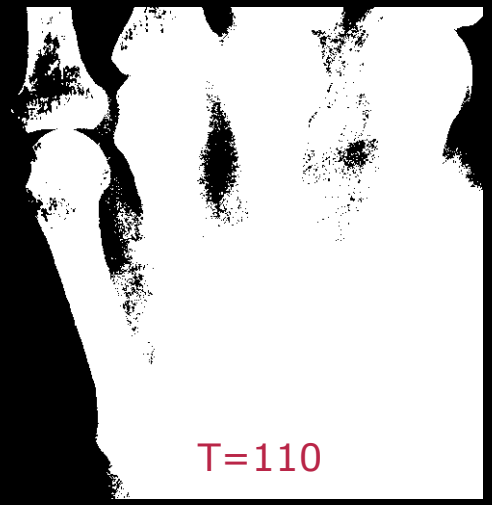
# Thresholding

- A threshold $T$ is a value
  - Pixels below that value is set to 0 (background)
  - Pixels equal or above is set to 1 (object)
- One threshold value for the entire image
  - Difficult to choose!

$$\text{if } f(x,y) \leq T \text{ then } g(x,y) = 0$$

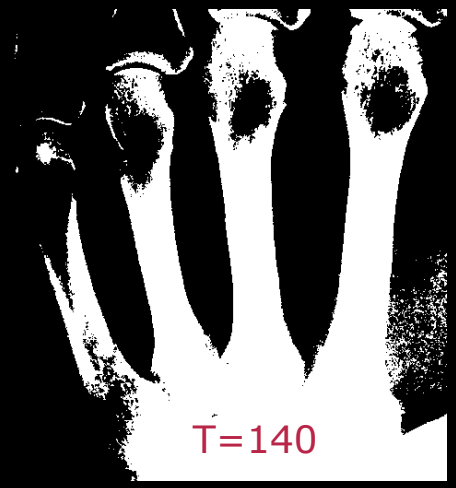
$$\text{if } f(x,y) > T \text{ then } g(x,y) = 255$$

# Thresholding

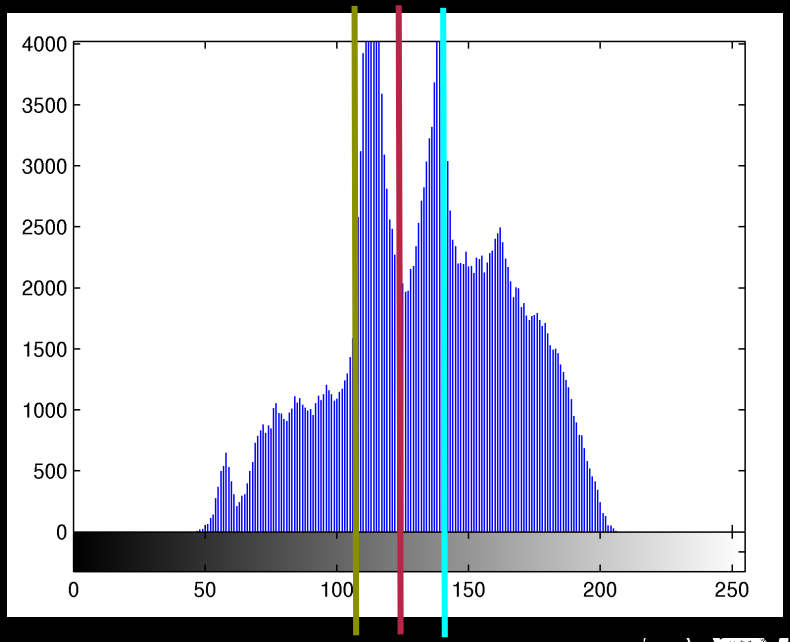




Background and bone have same value!



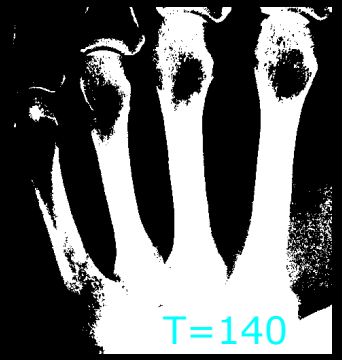
# Thresholding based on the histogram



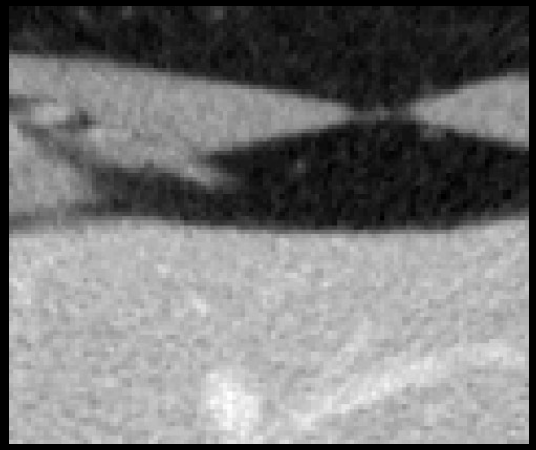

The bones are visible in the histogram!

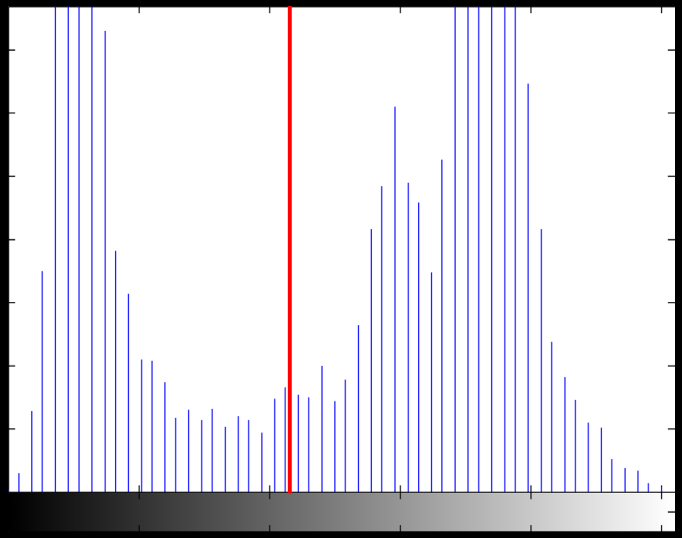
But mixed with soft-tissue





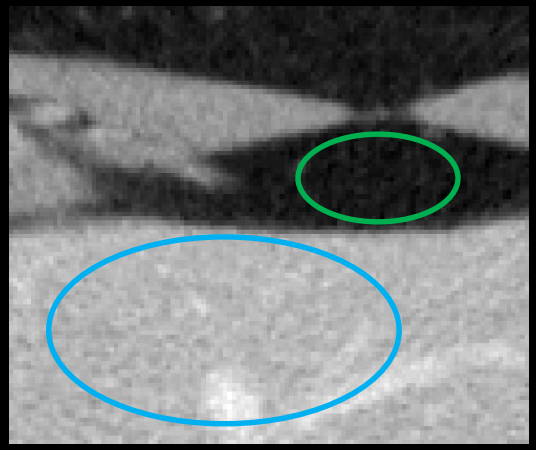
# Automatic Tresholding

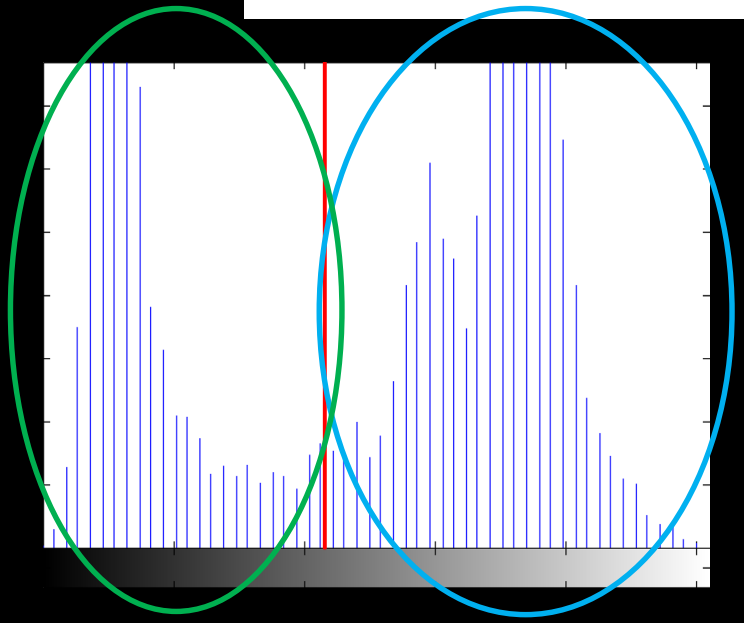




Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." IEEE transactions on systems, man, and cybernetics 9.1 (1979): 62-66.

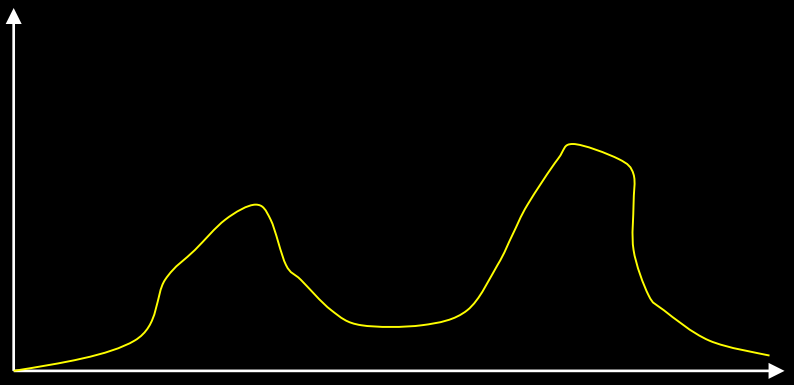
# Automatic Tresholding Otsu's method

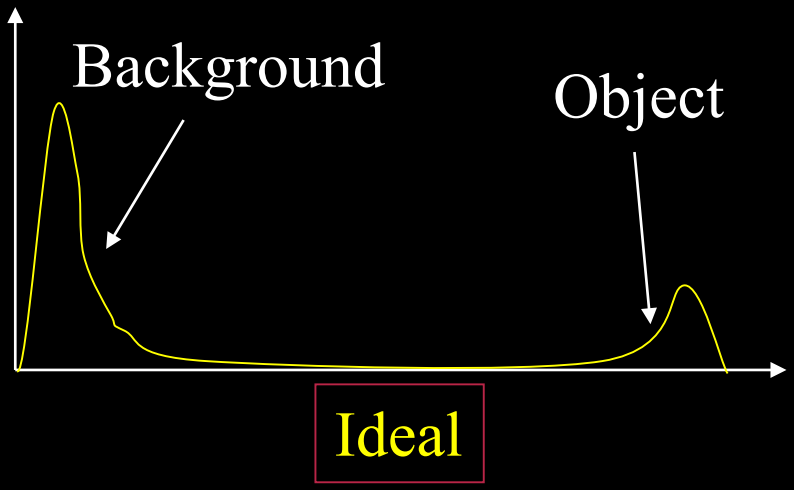




- Two classes: background and object
- T divides pixels into object and background
- Compute pixel value variance in each class
- Find T that minimises combined variance

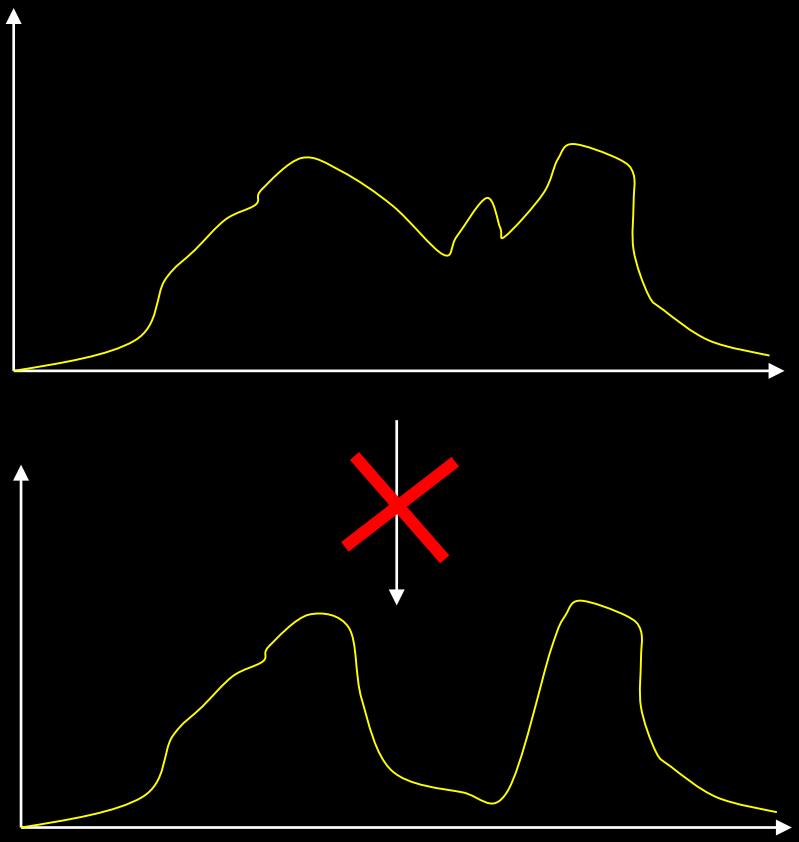
# Segmentation – histogram shaping





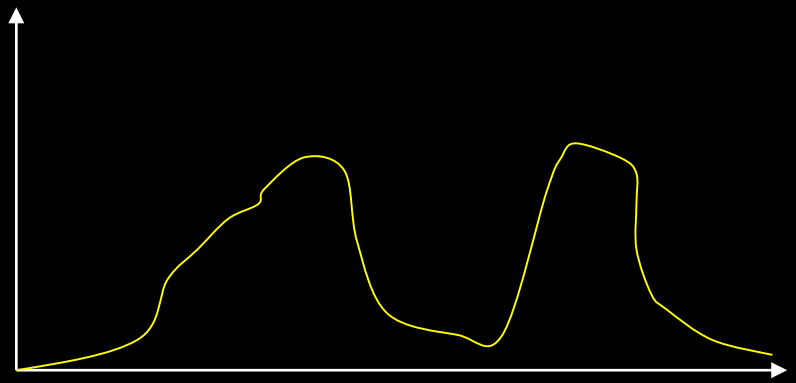

- With a threshold you want a histogram with two peaks
  - *Bimodal*
- An ideal histogram has well separated peaks
- Obtaining a bi-modal histogram is very important in the image acquisition

# Histogram shaping

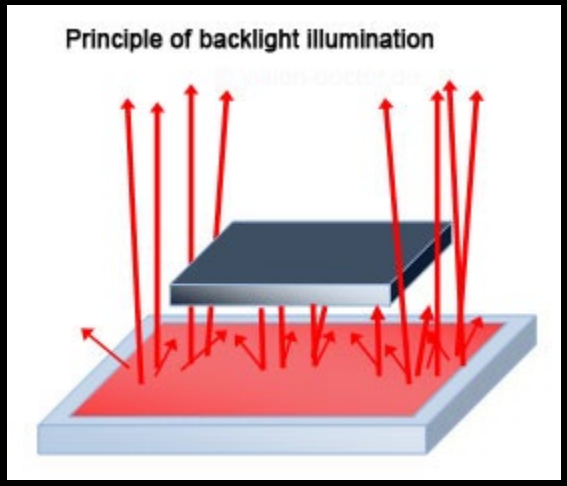


- It is not possible to "unmix" using gray level transformations

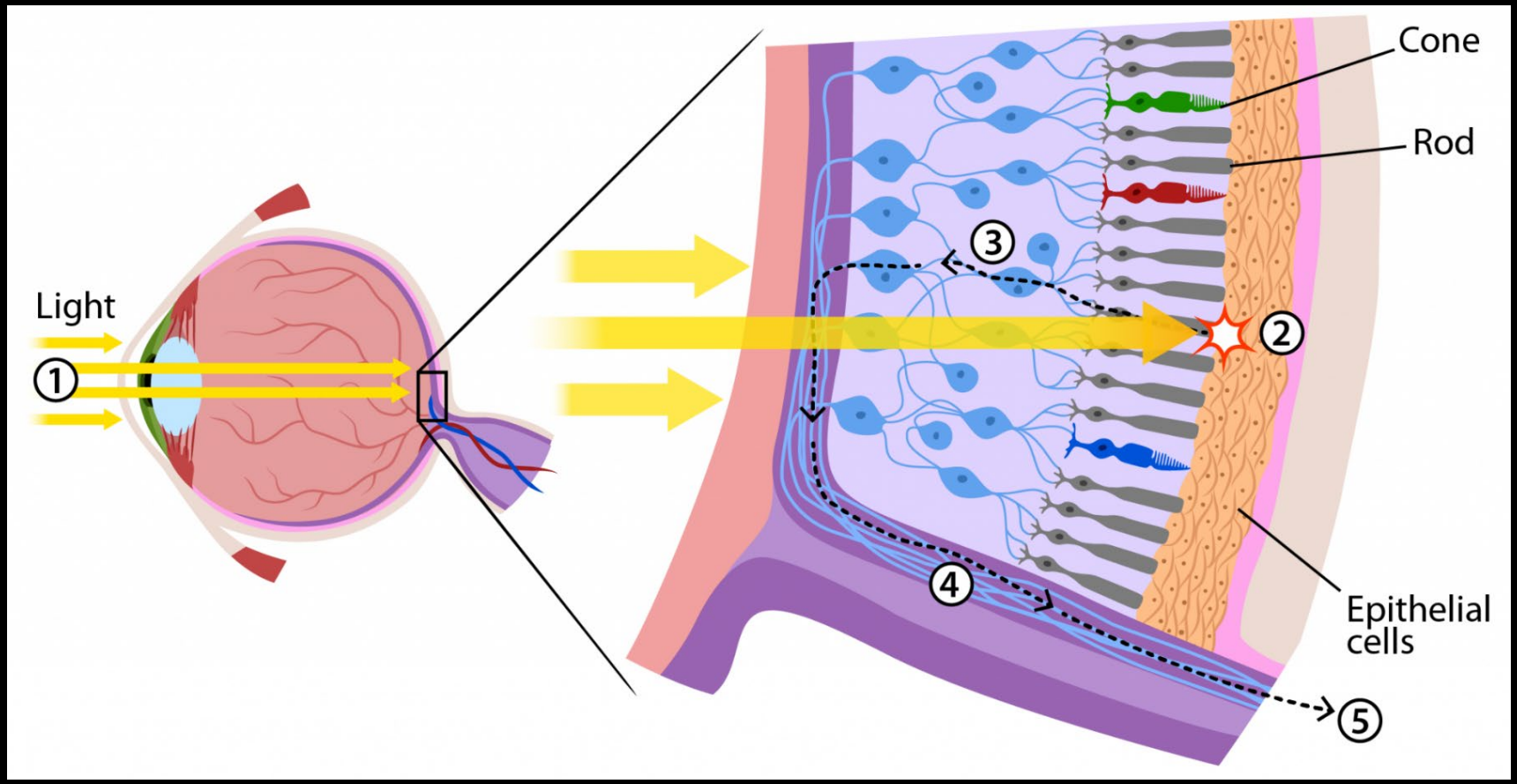
# How to obtain good histograms



- With cameras
  - Light
  - Setup
  - Camera
  - Lens
  - Backlight



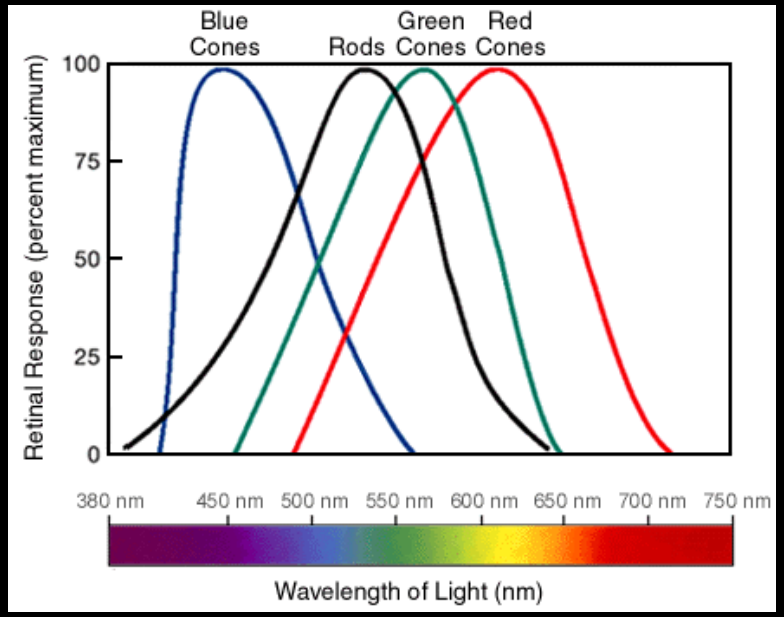
# Colour images and colour perception
# The Human Eye




https://askabiologist.asu.edu/rods-and-cones

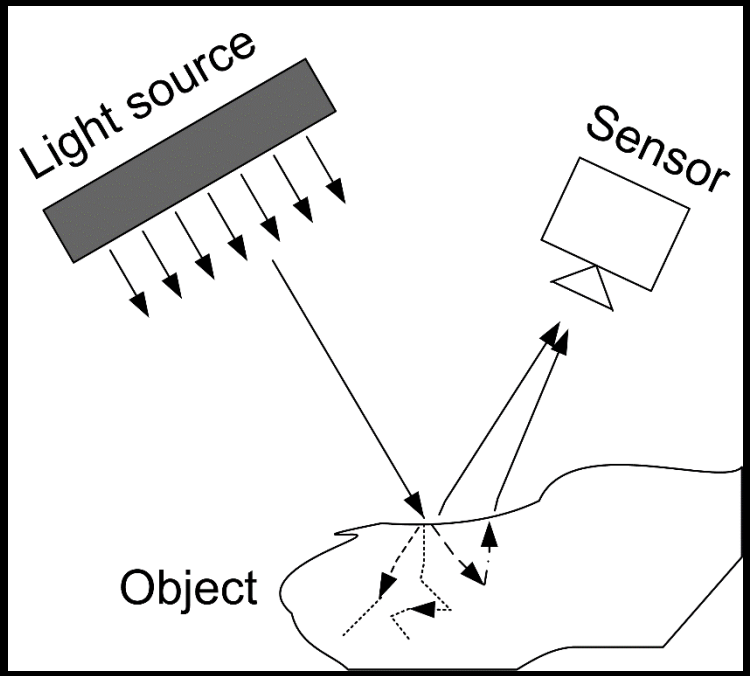
# Color sensitivity

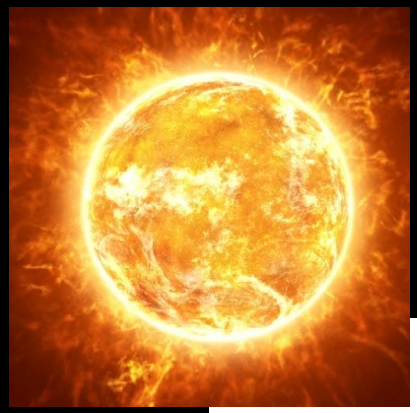
| Photoreceptor cell | Wavelength in nanometers (nm) | Peak response in nanometer (nm) | Interpretation by the human brain |
|---|---|---|---|
| Cones (type L) | [400-680] | 564 | Red |
| Cones (type M) | [400-650] | 534 | Green |
| Cones (type S) | [370-530] | 420 | Blue |
| Rods | [400-600] | 498 | Shade of gray |

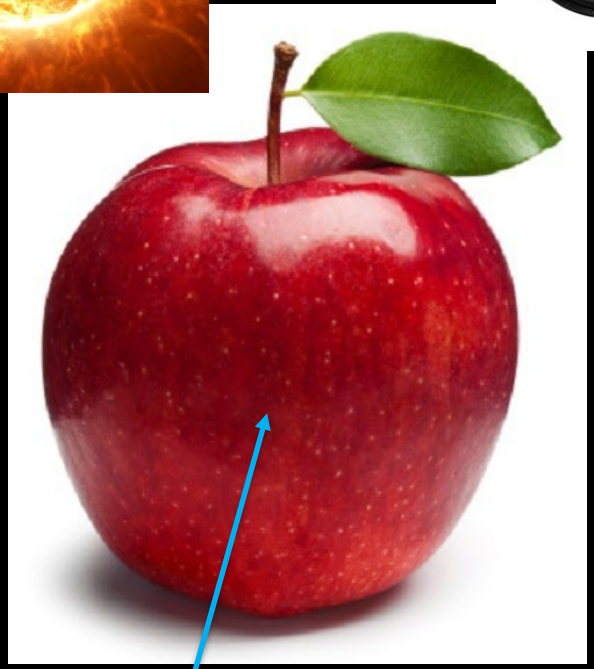



https://askabiologist.asu.edu/rods-and-cones

# Object colors
## Subtractive colors





All other colors than red absorbed

# Object colors
## Additive colors

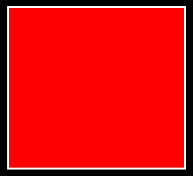
- Additive colours: Final colour is made by mixing red, green, and blue
- RGB = Red, Green, and Blue
- Television, computers, digital cameras use the “RGB color space”
- Typically the values of R, G, and B lie between 0 and 255

# RGB Colours

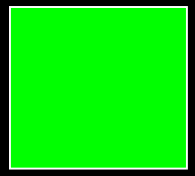
RGB = (0,0,0)

RGB = (255,255,255)

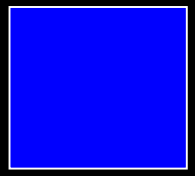
- When alle three “Lamps” are turned of we get black
- When all three “lamps” are on what do we get?



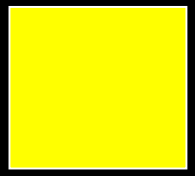
(255,0,0)



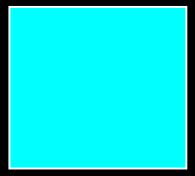
(0,255,0)



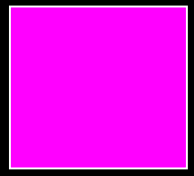
(0,0,255)



(255,255,0)

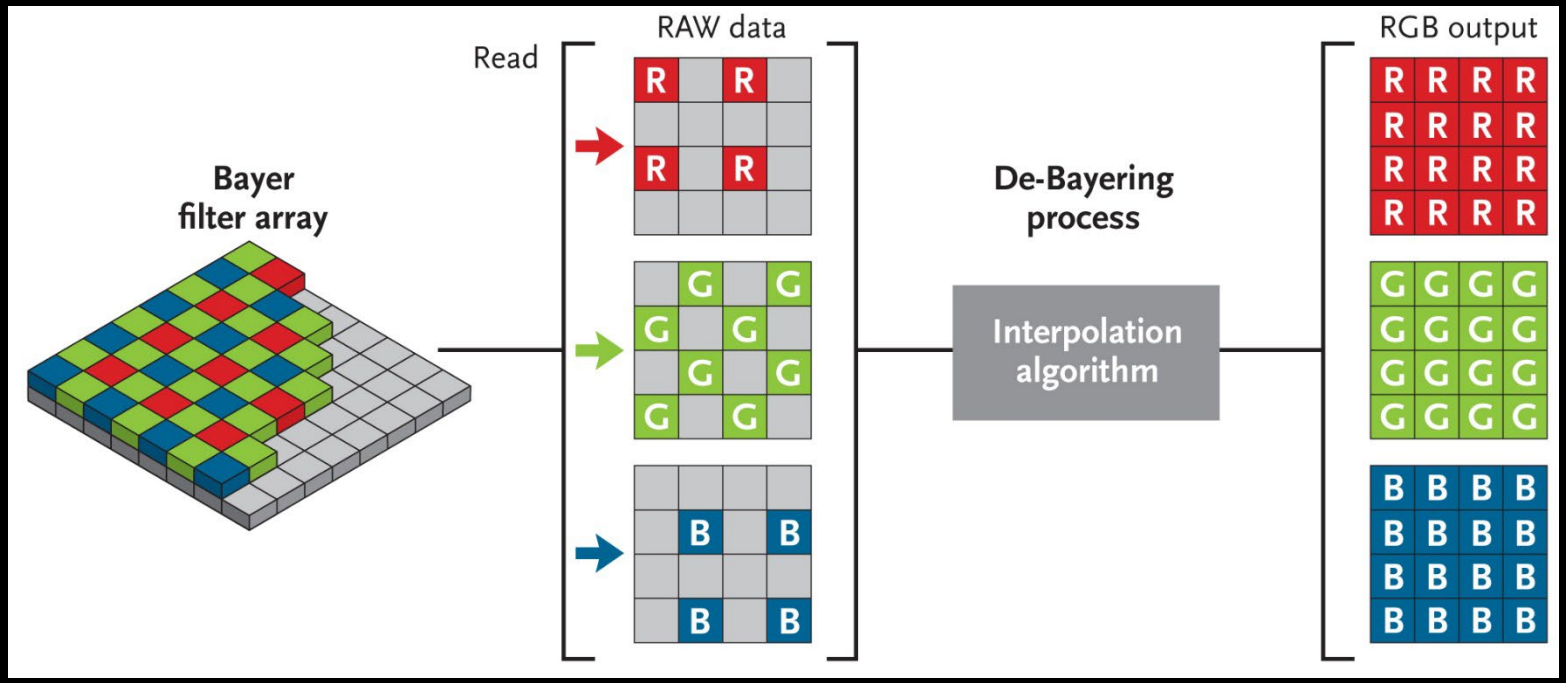


(0,255,255)



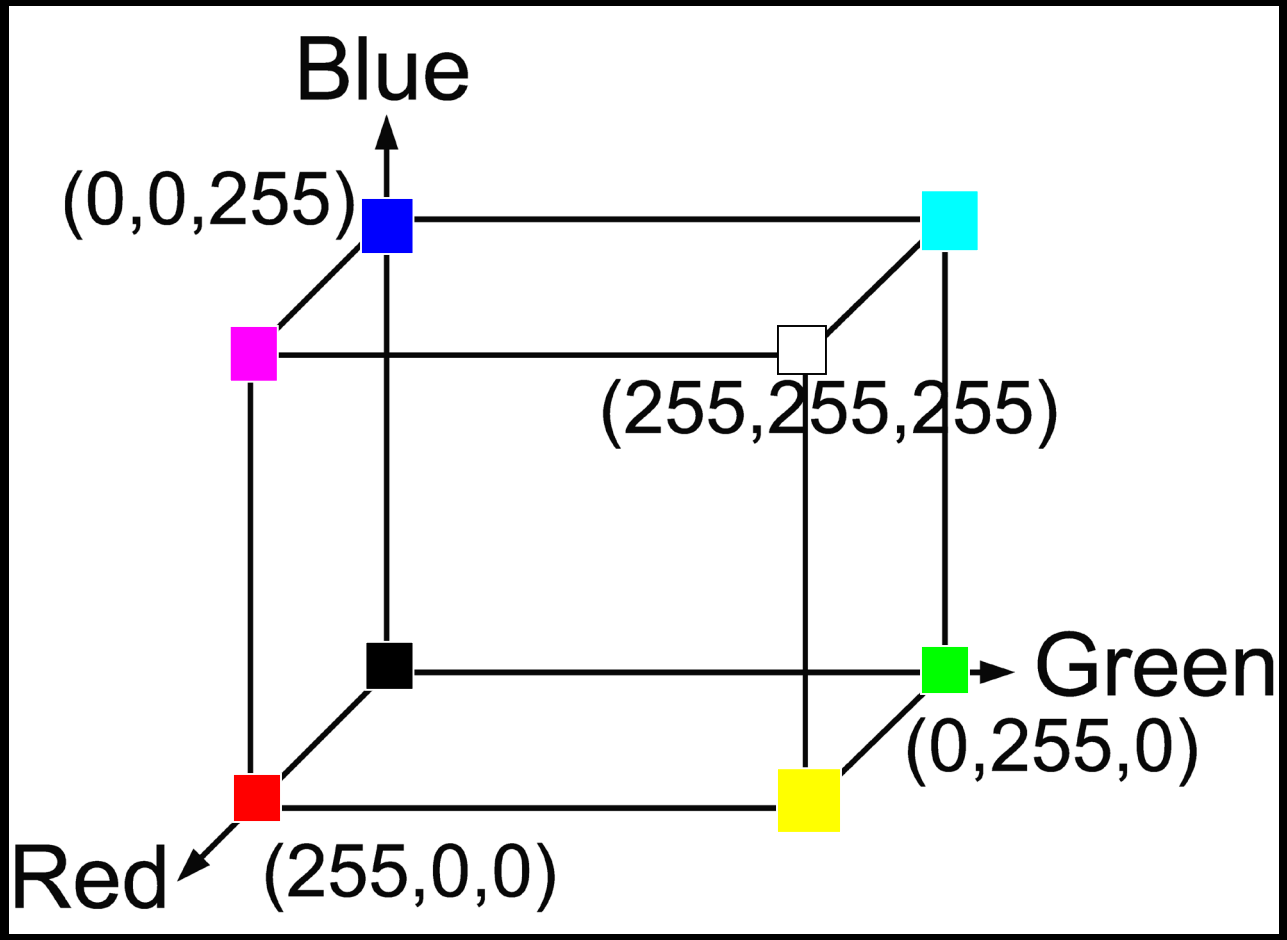
(255,0,255)

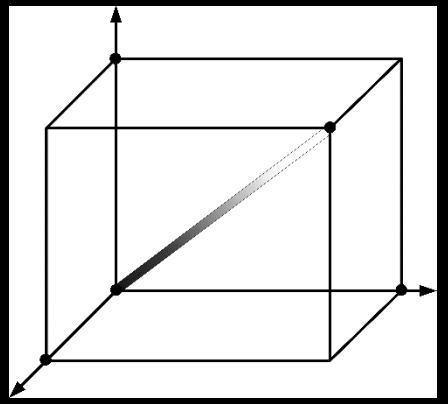
# Color camera with one sensor




http://www.skyandtelescope.com/astronomy-resources/astrophotography-tips/redeeming-color-planetary-cameras/

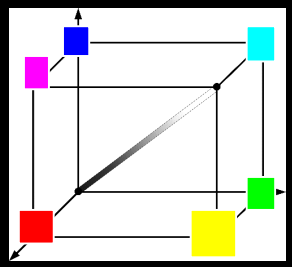
# RGB color space



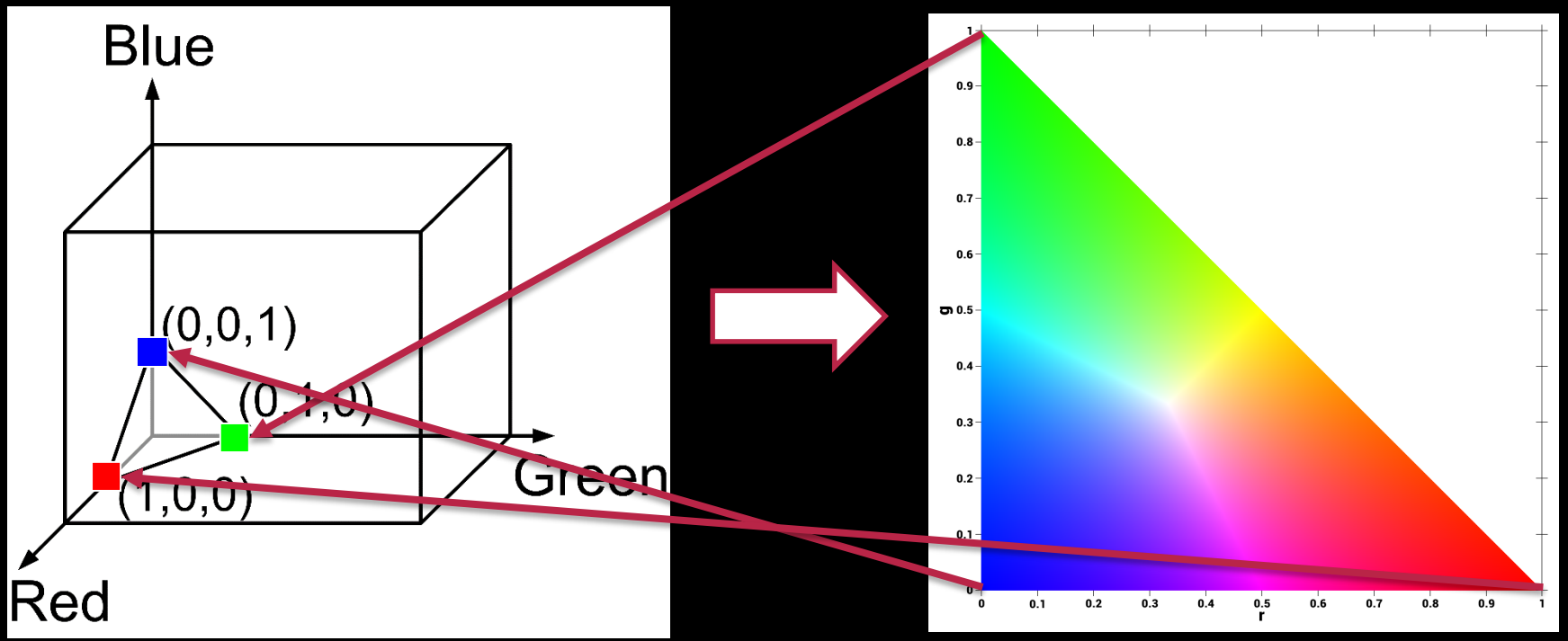
# Converting colour to grayscale

$$v = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

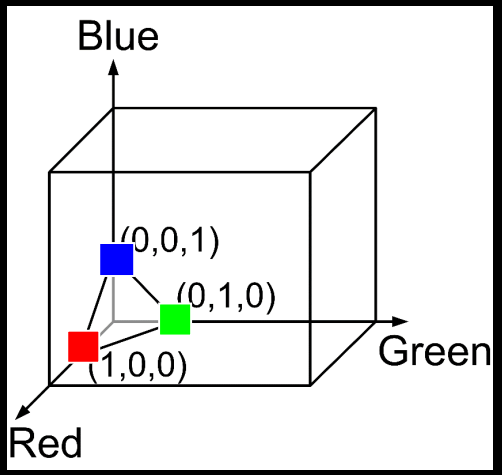


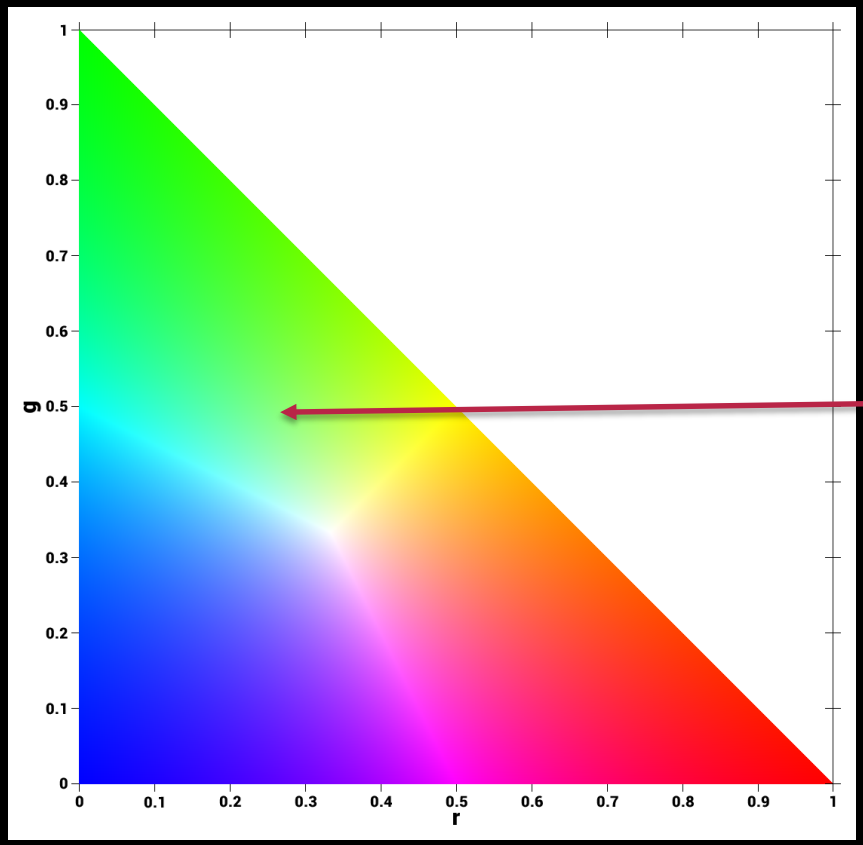
# Normalized RGB colors

$$(r,g,b) = \left(\frac{R}{R+G+B}, \frac{G}{R+G+B}, \frac{B}{R+G+B}\right)$$



# Another RGB representation



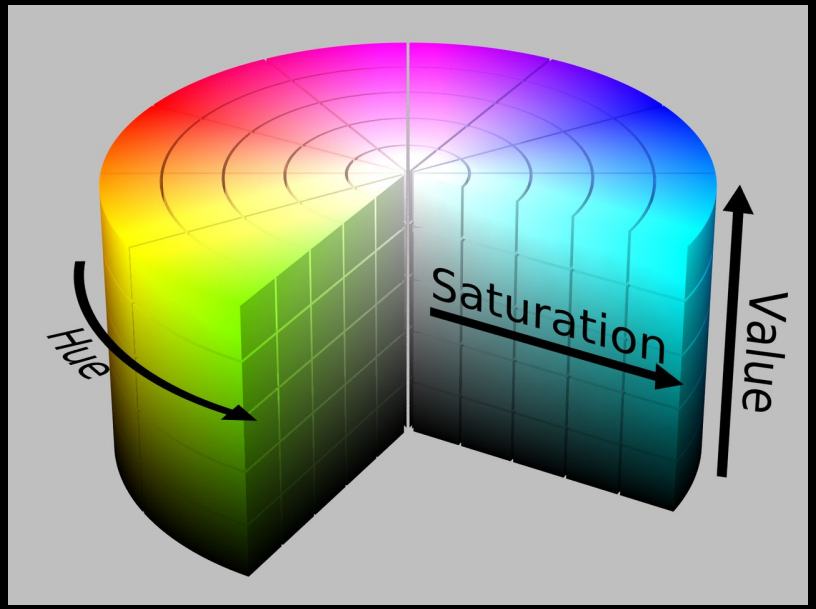




(r,g,I)

$$I = \frac{R+G+B}{3}.$$

# HSI Color Reprentation

- Hue – the dominant wave length in the perceived light (the pure color)
- Saturation – the purity of the color
- Intensity – the brightness of the color (sometimes called the value)



# Converting between RGB and HSI

- You have an RGB value
- You want the corresponding HSI value

$$H = \begin{cases} \cos^{-1}\left(1/2 \cdot \frac{(R-G)+(R-B)}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}}\right), & \text{if } G \geq B; \\ 360^\circ - \cos^{-1}\left(1/2 \cdot \frac{(R-G)+(R-B)}{\sqrt{(R-G)(R-G)+(R-B)(G-B)}}\right), & \text{Otherwise.} \end{cases} \quad (8.8)$$

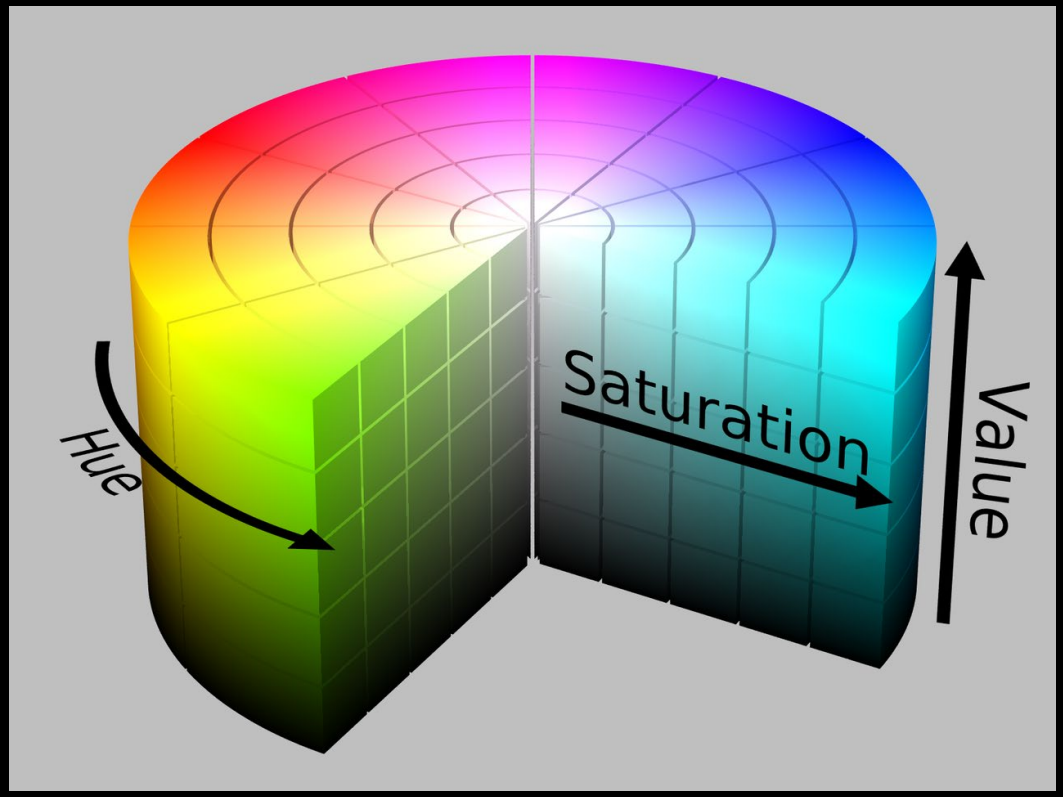
$$H \in [0, 360[$$

$$S = 1 - 3 \cdot \frac{\min\{R, G, B\}}{R+G+B} \qquad S \in [0, 1] \quad (8.9)$$

$$I = \frac{R+G+B}{3} \qquad I \in [0, 255] \ , \quad (8.10)$$

# Why other colorspaces

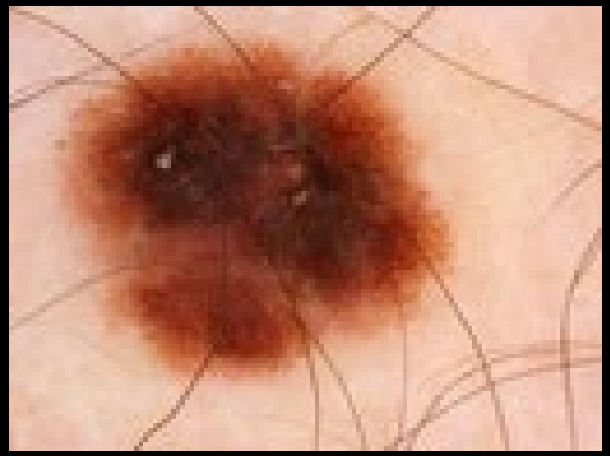
- Why should we use for example HSI ?



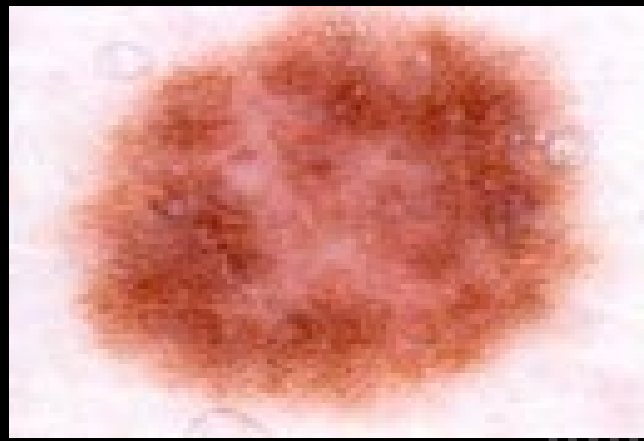
# Melanoma segmentation

- An algorithm that can do pixelwise classification
  - Background / skin
  - Melanoma
- Use the colors

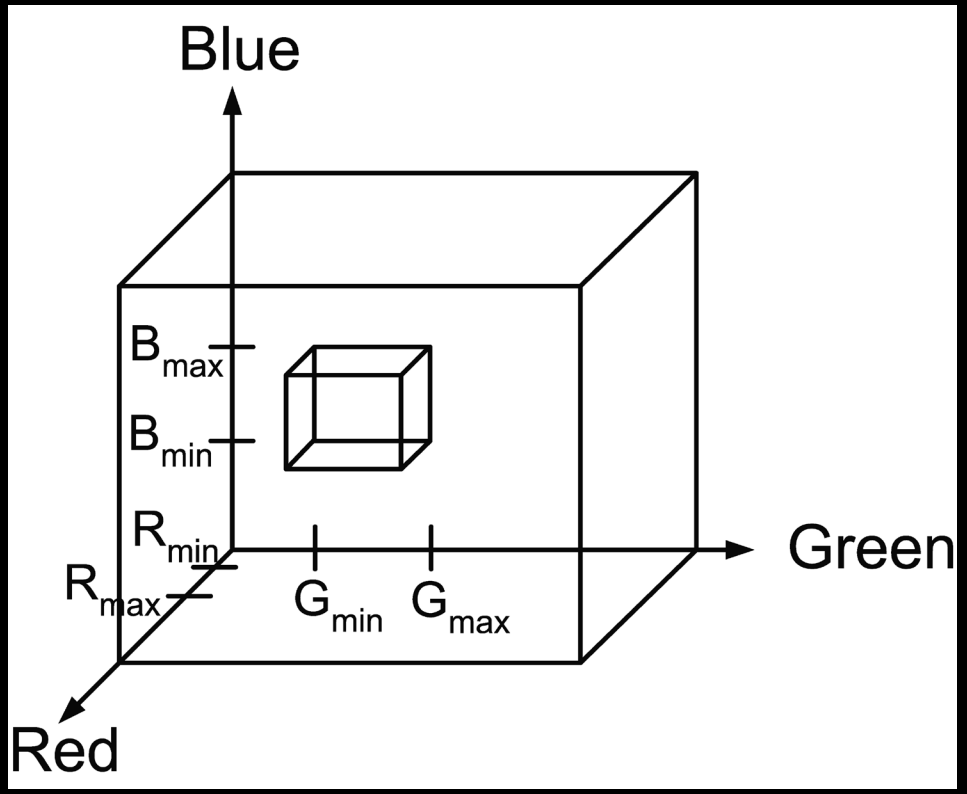
# Melanoma segmentation – color variation





# Color thresholding



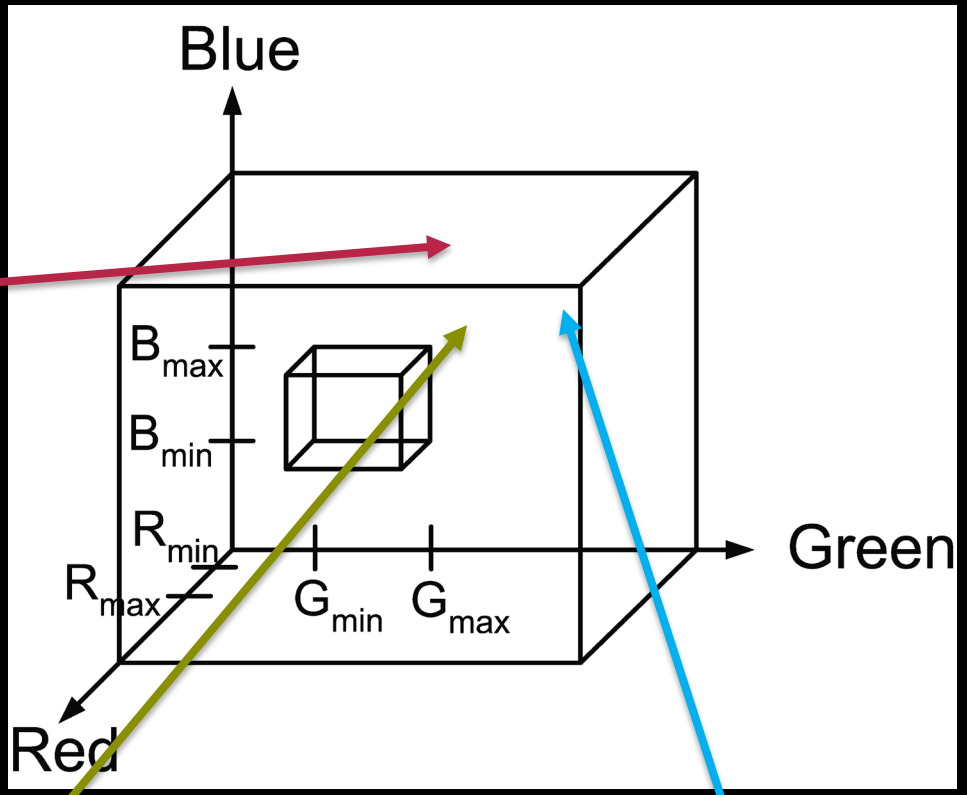

If

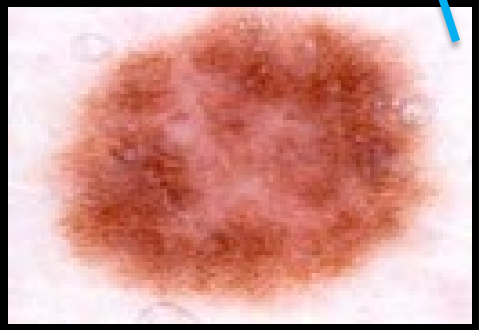
$R > R_{min}$ and $R < R_{max}$ and
$G > G_{min}$ and $G < G_{max}$ and
$B > B_{min}$ and $B < B_{max}$

Then $g(x, y) = 255$

Else $g(x, y) = 0$

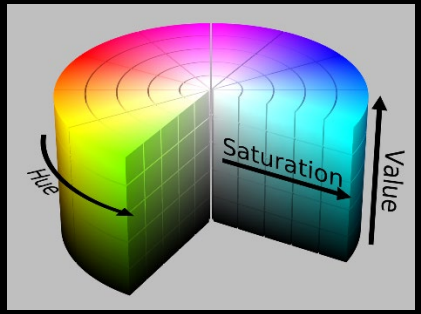
# Color thresholding

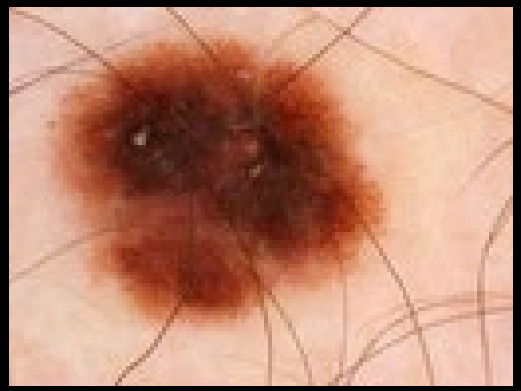


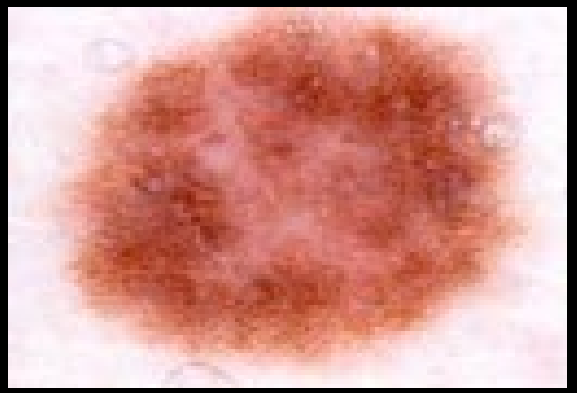



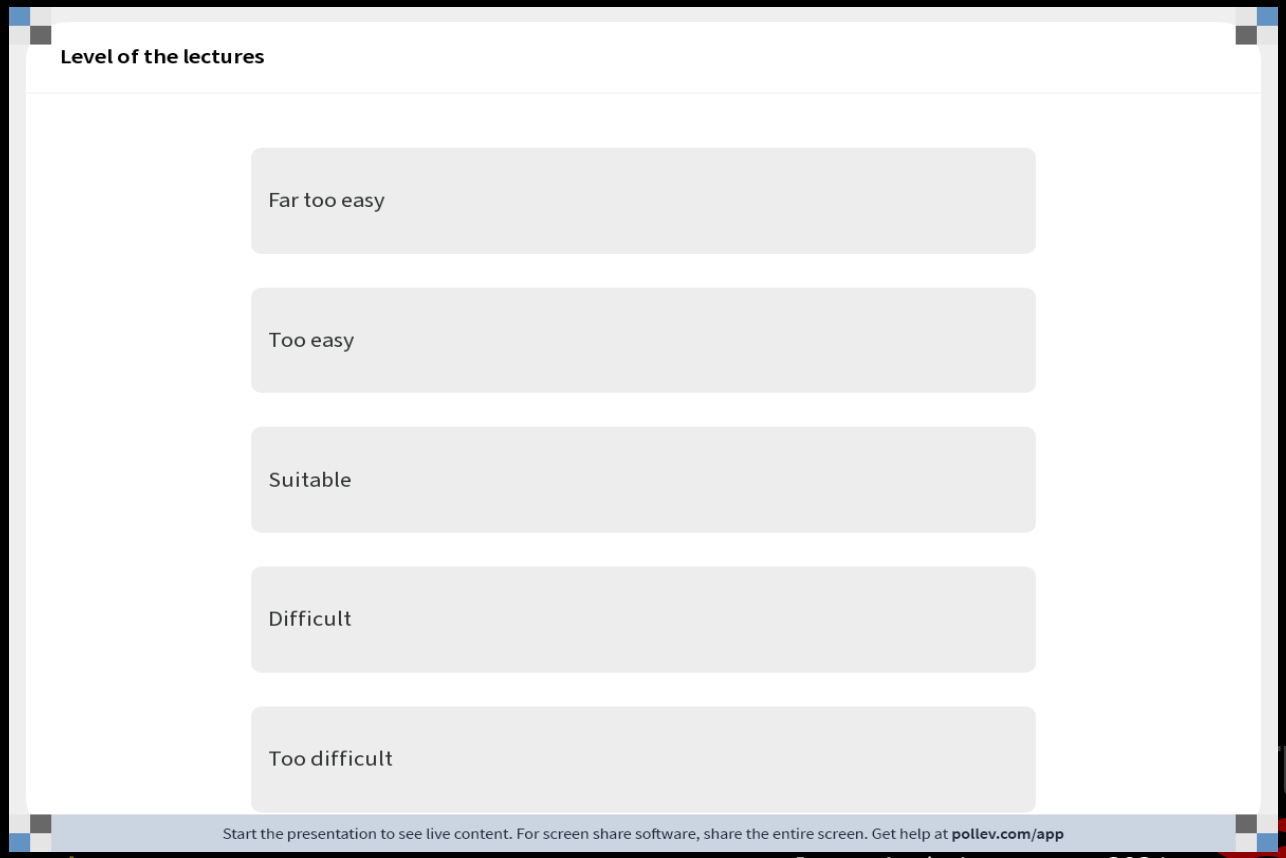
# Color variation

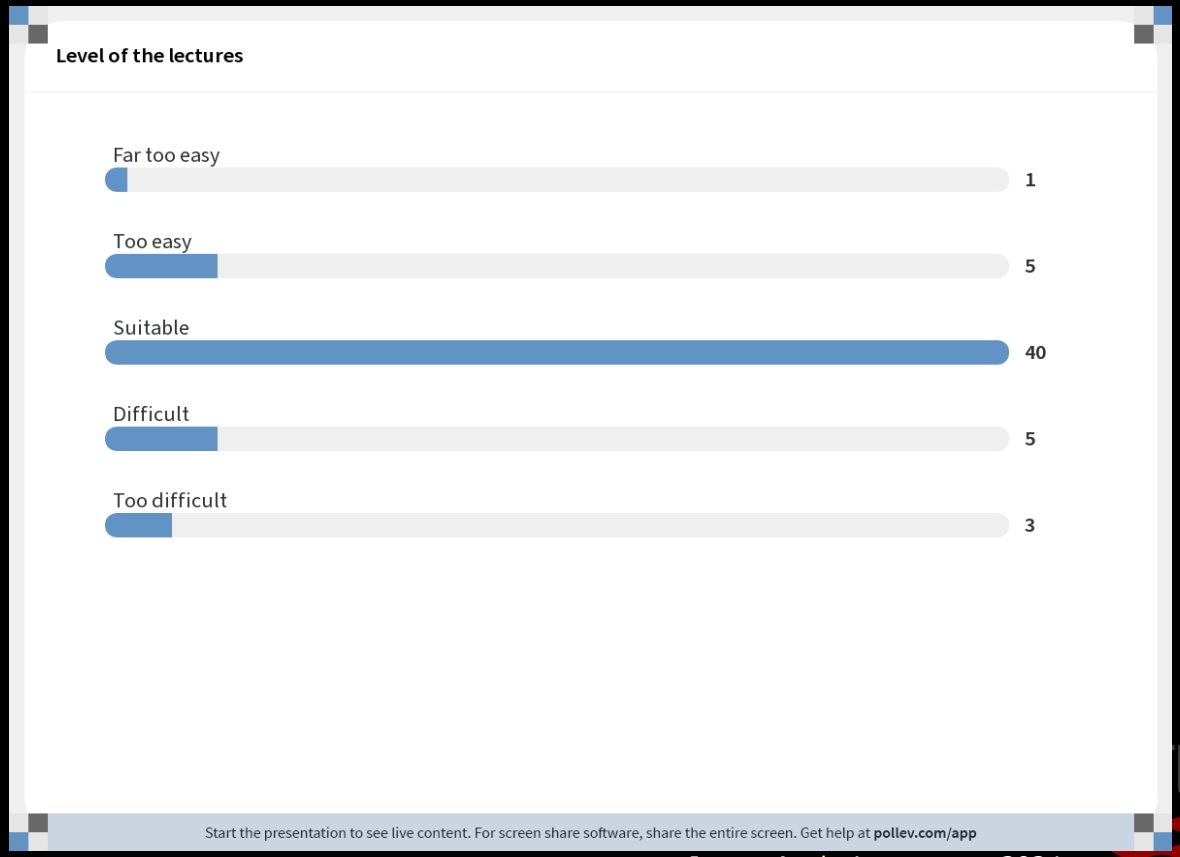
- The major variation is in the brightness
  - This will spread out the values in RGB space
- The Hue is rather constant
- HSI Space
  - HUE and saturation rather stable
  - Only variation in intensity / value

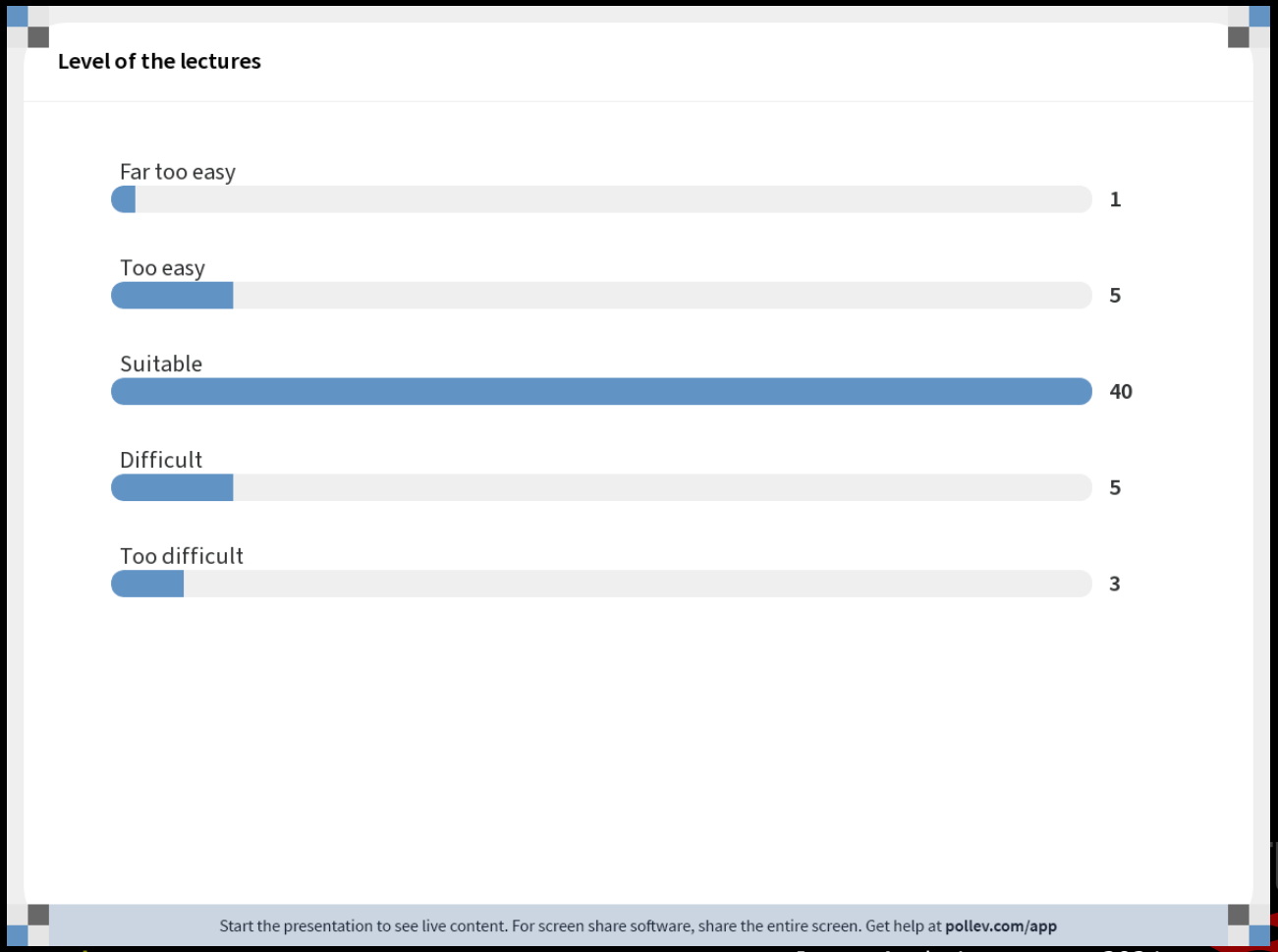


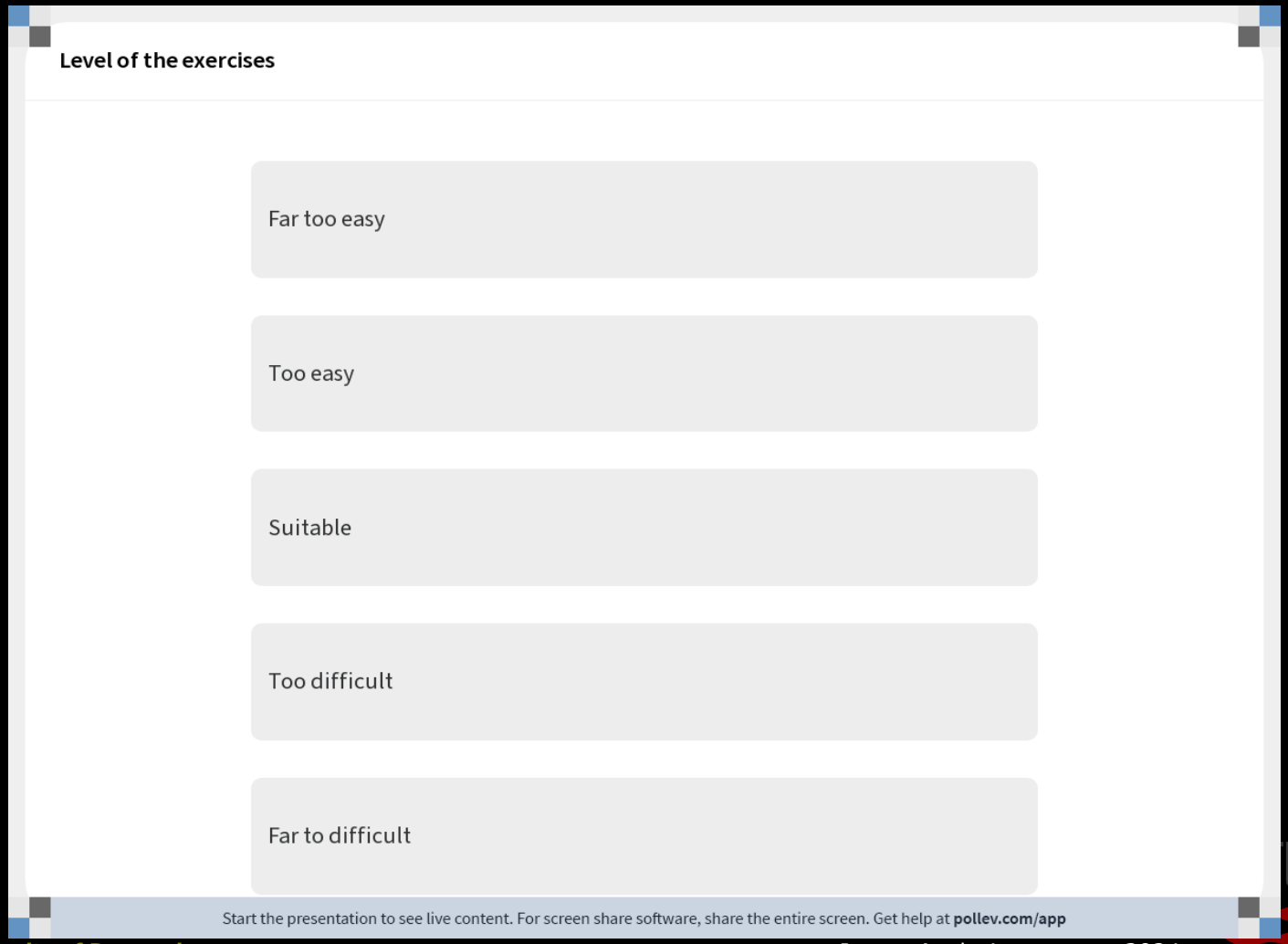



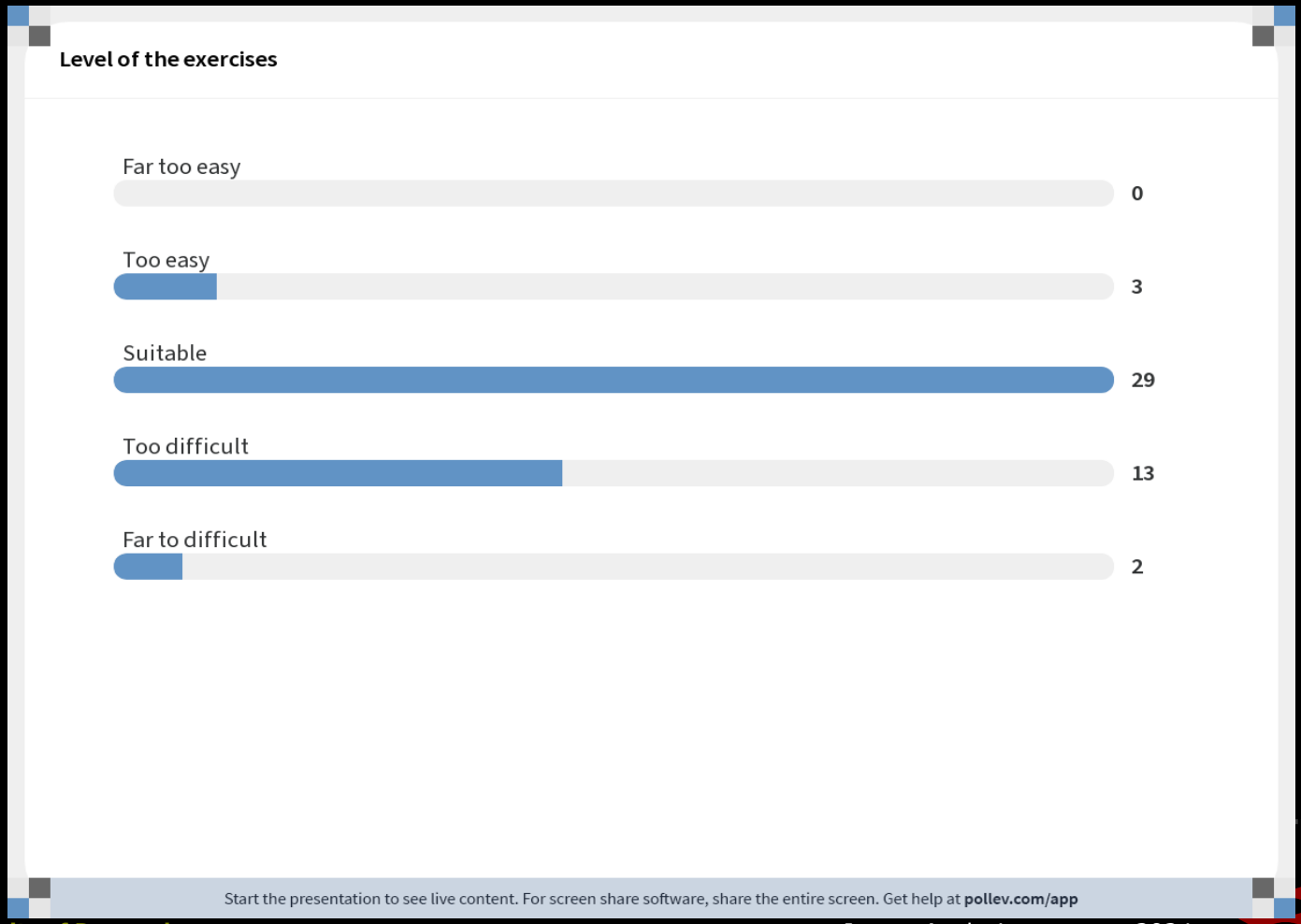


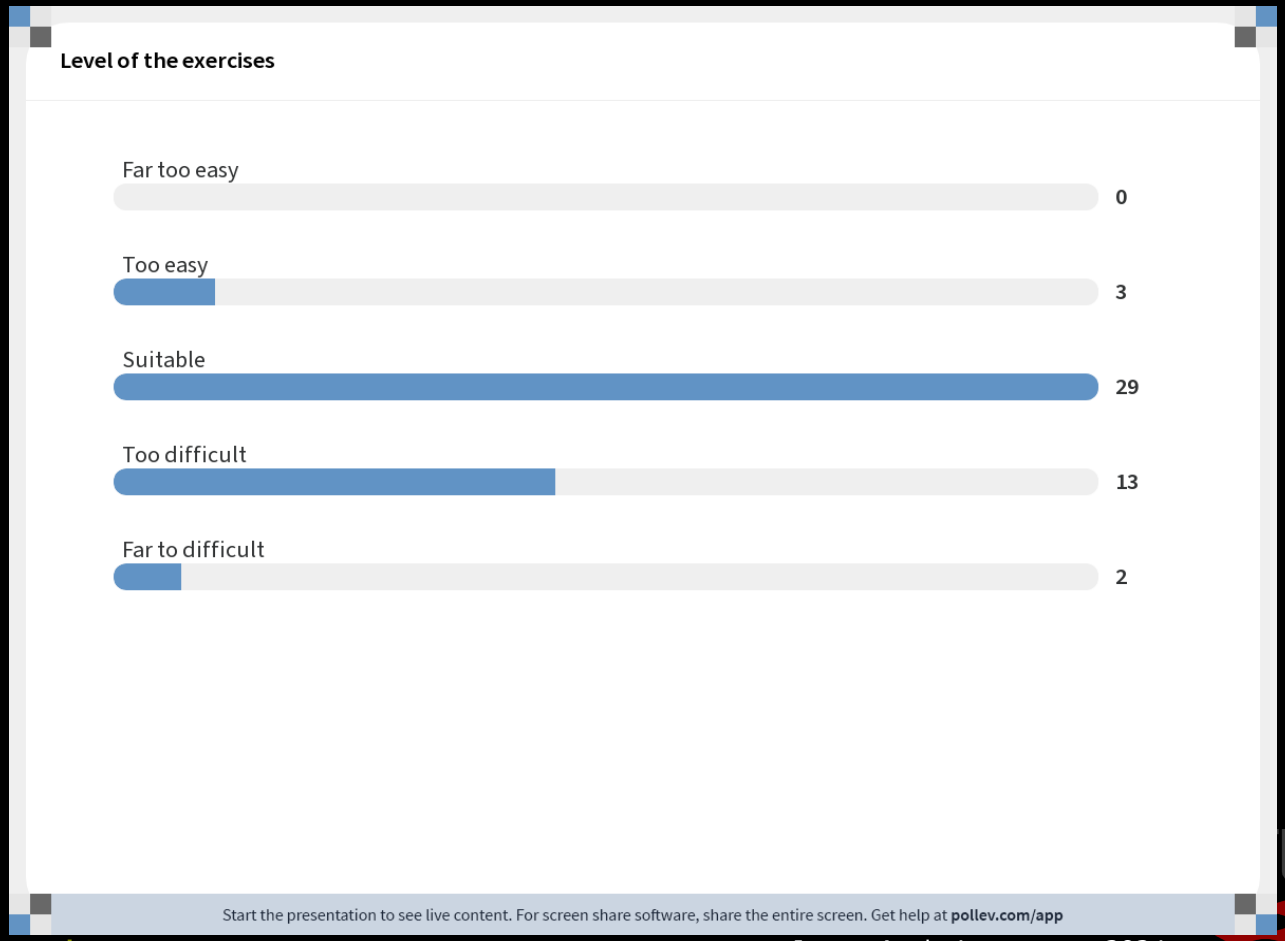

Level of the lectures
Far too easy
Too easy
Suitable
Difficult
Too difficult
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Level of the lectures
Far too easy
1
Too easy
5
Suitable
40
Difficult
5
Too difficult
3
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Level of the lectures
Far too easy
1
Too easy
5
Suitable
40
Difficult
5
Too difficult
3
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Level of the exercises
Far too easy
Too easy
Suitable
Too difficult
Far to difficult
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Level of the exercises
Far too easy
0
Too easy
3
Suitable
29
Too difficult
13
Far to difficult
2
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app


Level of the exercises
Far too easy
0
Too easy
3
Suitable
29
Too difficult
13
Far to difficult
2
Start the presentation to see live content. For screen share software, share the entire screen. Get help at pollev.com/app

# Next week

- Neighbourhood processing (Filtering)
- Morphology