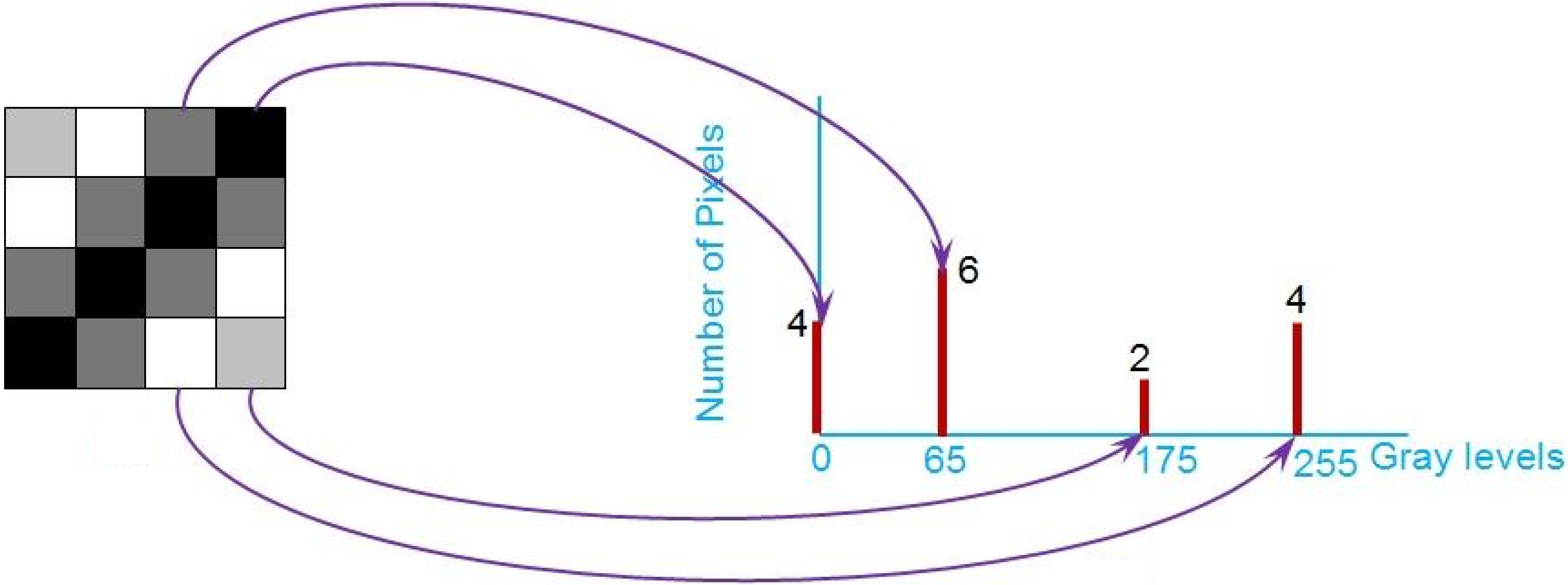
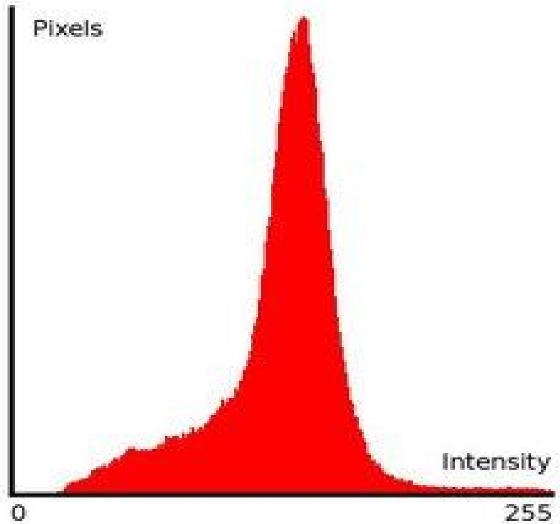
# Image comparison based Histograms

# Image and its histogram







# Histogram and PDF

- Image histogram normalized so that the total area under the histogram is 1
- Count the brightness levels {0, 1, 2, ..,255}
- We have h(1), h(2), ...h(255)
- Image ≡ (i x j) matrix
- Total pixels (N) = i x j
- p(1) = h(1)/N, p(2) = h(2)/N, ...,p(255)=h(255)/N

# Colour image

- Three matrices
- R matrix, G matrix and B matrix
- (i x j) gray scale image has total pixes=i x j
- (i x j) colour image has total pixes=3.(i x j)

# Colour histogram

| Red | Green | Blue |
|---|---|---|
| Bin0 | Bin0 | Bin0 |
| Bin0 | Bin0 | Bin1 |
| Bin0 | Bin0 | Bin2 |
| Bin0 | Bin0 | Bin3 |
| Bin0 | Bin1 | Bin0 |
| ... | | |
| ... | | |
| ... | | |
| Bin3 | Bin3 | Bin2 |
| Bin3 | Bin3 | Bin3 |

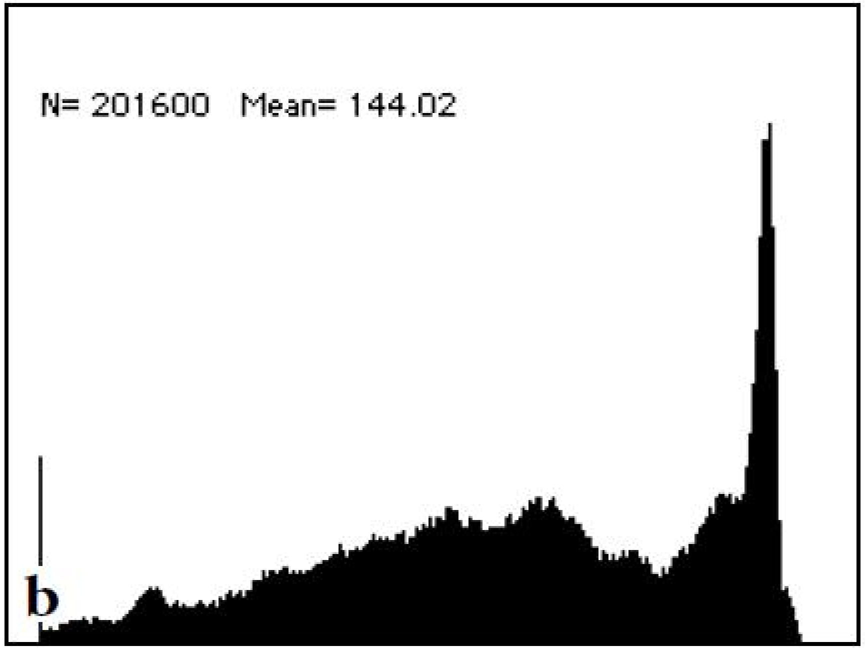
- Bin 0 is 0-63
- Bin 1 is 64-127
- Bin 1 is 128-191
- Bin 1 is 192-255

Total combinations = 4 x 4 x 4 = 64

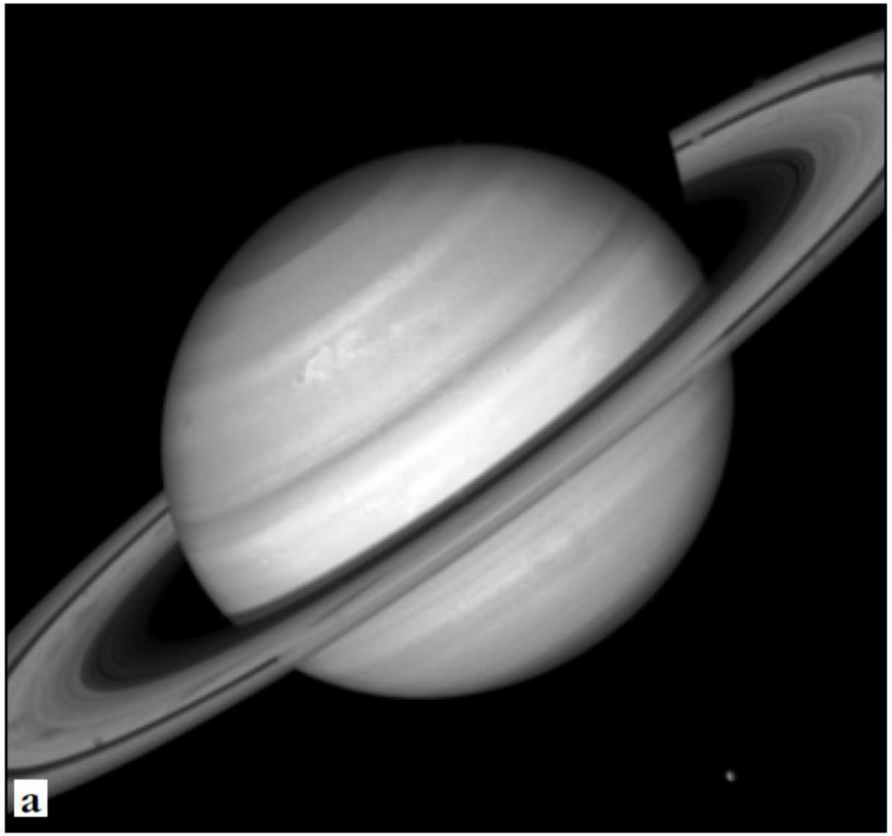
# Histogram and entropy

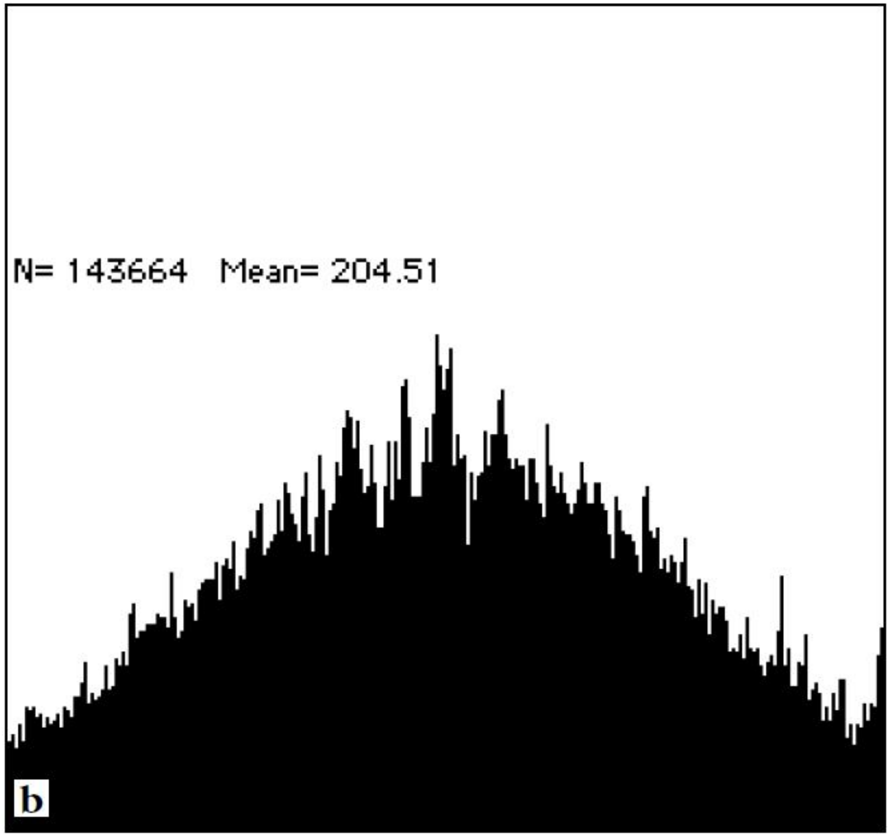
$$H = -\sum_{i=0}^{255} p_i . \log(p_i)$$

# Girl image and its histogram

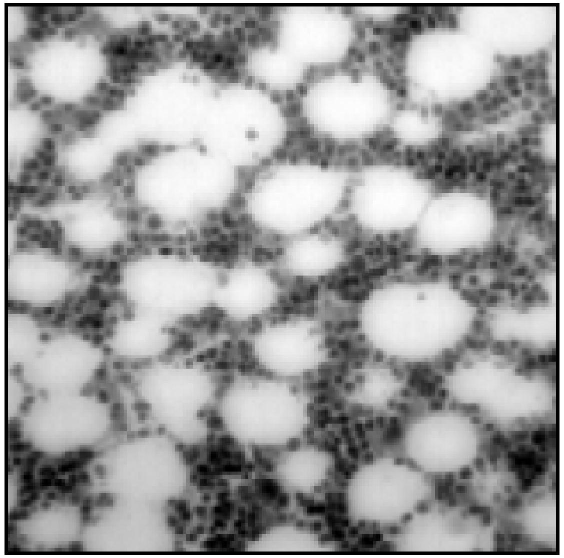



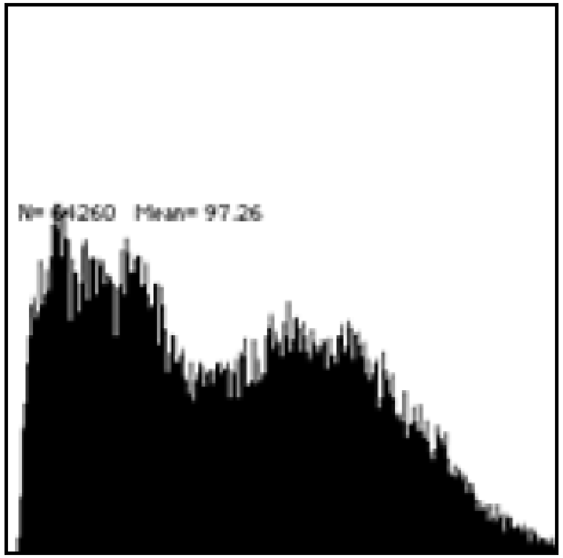
# Saturn image and its histogram

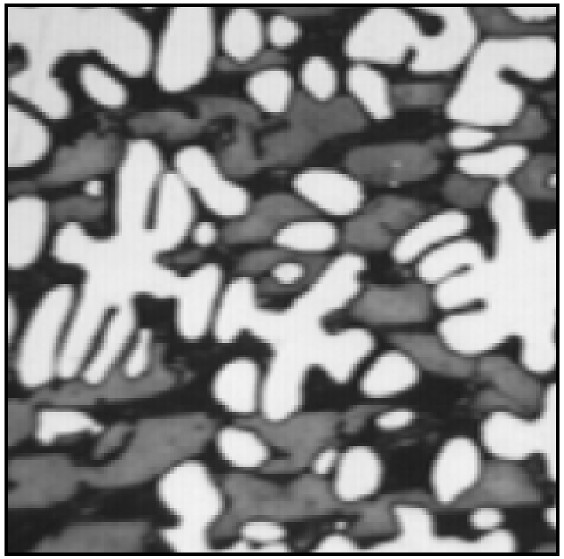


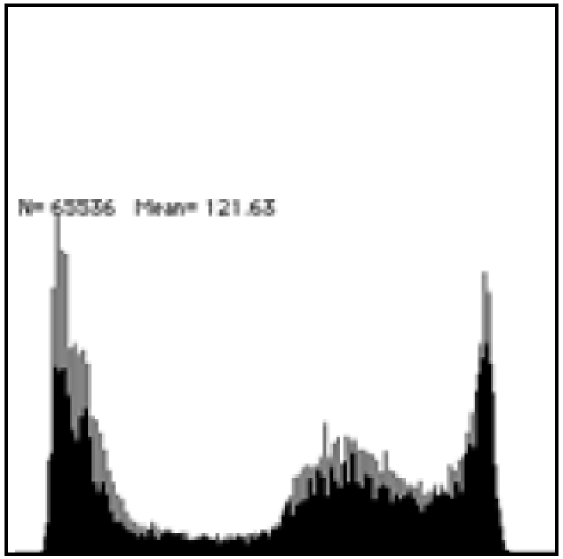



# Bone marrow & dendrites - histograms

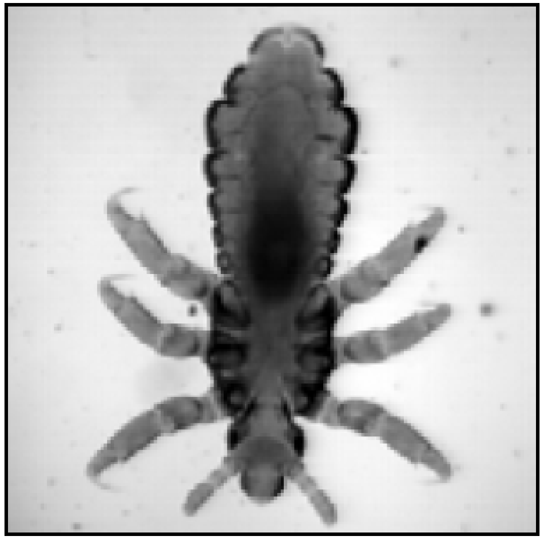


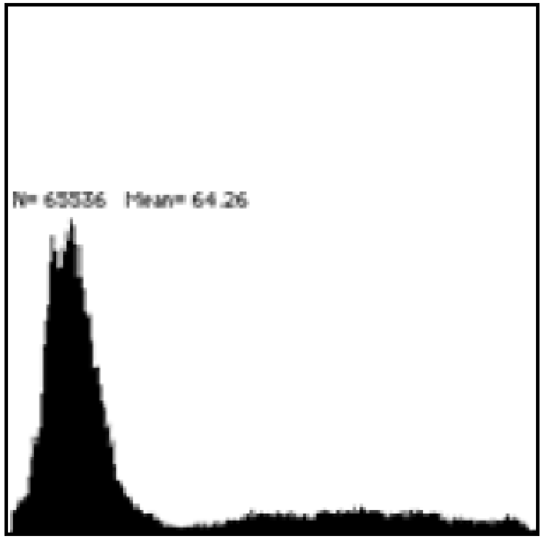


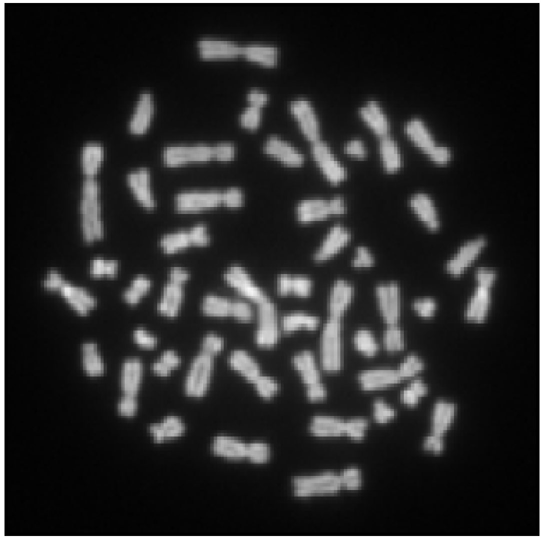





# Bug & chromosome - histograms





# Entropy comparison

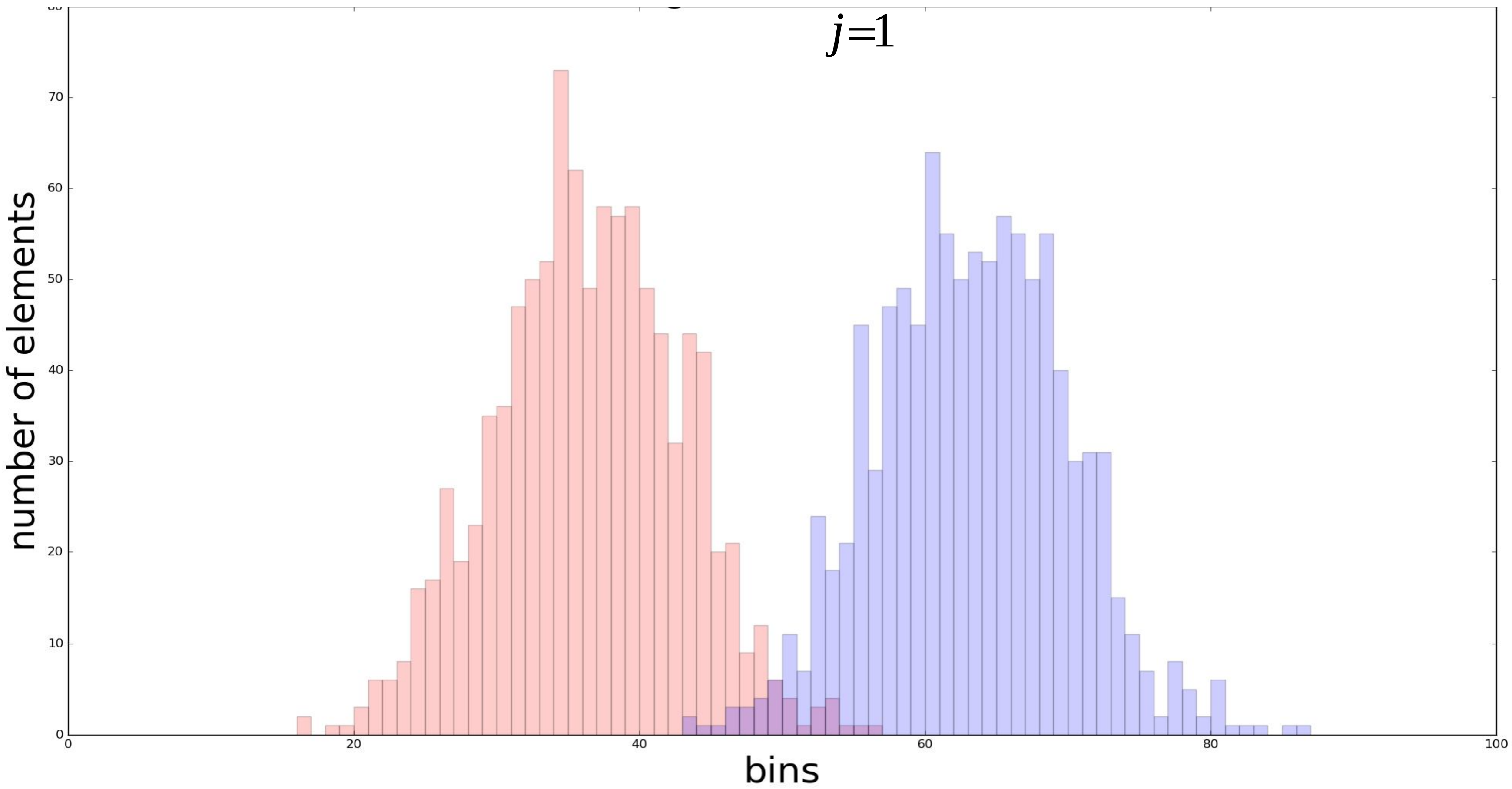
| Image | H |
|---|---|
| girl | 7.538 |
| Saturn | 4.114 |
| bone marrow | 7.780 |
| dendrites | 7.415 |
| bug | 6.929 |
| chromosomes | 5.836 |

# Why should we compute entropy?

- Recall source coding theorem
- Average code length >=entropy
- Efficient storage of images

# Histogram intersection algorithm

$$\mathrm{intersection} = \sum_{j=1}^{n} \min\left(I_j, M_j\right)$$



# Explanation of histogram intersection

- Input image: I
- Model image: M
- Both have N bins
- Take the first bin of I and first bin of M
- Compare their frequencies
- Which ever is minimum noted down
- Do it for second bin, third bin... up to $N^{th}$ bin
- Create a new histogram

# Histogram intersection for classification

- One input image (I)
- Say K model images ($M_1$, $M_2$, ...,$M_K$)
- All have N bins
- Generate K intersection histograms
- A = Find the area under intersection = quantify the intersection = count the number of elements of each bin of intersection histogram
- Now we have $A_1$, $A_2$, ..., $A_K$
- Choose the maximum out of $A_1$, $A_2$, ..., $A_K$

# Algorithm

*Given model image set = {$M_1$, $M_2$, $M_3$..., $M_k$}*

*Given test image = I*

*For n=1:k*

*Read test image (I)*

*Read model image $M_k$*

*Generate two matrices corresponding to image I & $M_k$*

*Generate corresponding histograms ($h_1$ & $h_2$)*

*Find intersetcion*

*Store intersection similarity $s_{I\_Mk}$*

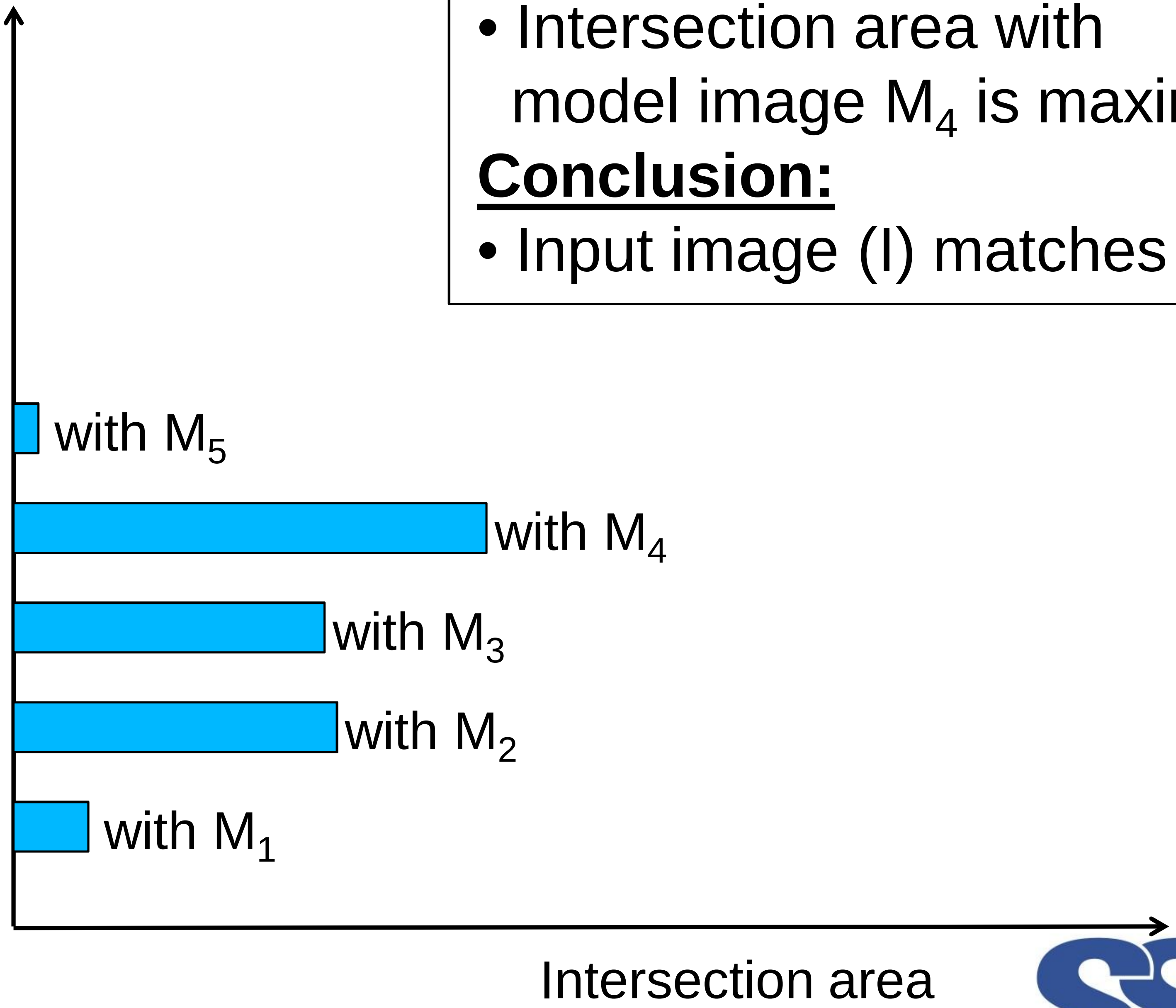
*End*

***Choose the image which has highest s***

- Intersection area with model image $M_4$ is maximum

**<u>Conclusion:</u>**

- Input image (I) matches with $M_4$



# Other distance measures

- Intersection captures **<u>similarity</u>** between two histograms
- Distance = (1 – similarity)
- Normalized histogram $\equiv$ PDF
- Comparing two histograms $\equiv$ comparing two PDFs

# How do you compare two PDFs?

- Relative entropy or KLD
- But KLD is not symmetric

$$D_{KL}(P \| Q) = -\sum_x P(x) \log \frac{Q(x)}{P(x)} \qquad D_{KL}(Q \| P) = -\sum_x Q(x) \log \frac{P(x)}{Q(x)}$$

$$= \sum_x P(x) \log \frac{P(x)}{Q(x)} \qquad = \sum_x Q(x) \log \frac{Q(x)}{P(x)}$$

But, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$ (in general)

$$\boxed{\sqrt{\frac{1}{2}[D_{KL}(P \| Q)]^2 + \frac{1}{2}[D_{KL}(Q \| P)]^2} \quad : J \ \ divergence}$$

# Algorithm

*Given model image set = {$M_1$, $M_2$, $M_3$..., $M_k$}*

*Given test image = I*

*for n=1:k*

*Read test image (I)*

*Read model image $M_k$*

*Generate two matrices corresponding to image I & $M_k$*

*Generate corresponding histograms ($h_1$ & $h_2$)*

*Normalize the histograms ($H_1$ & $H_2$)*

*They become PDF (P and Q)*

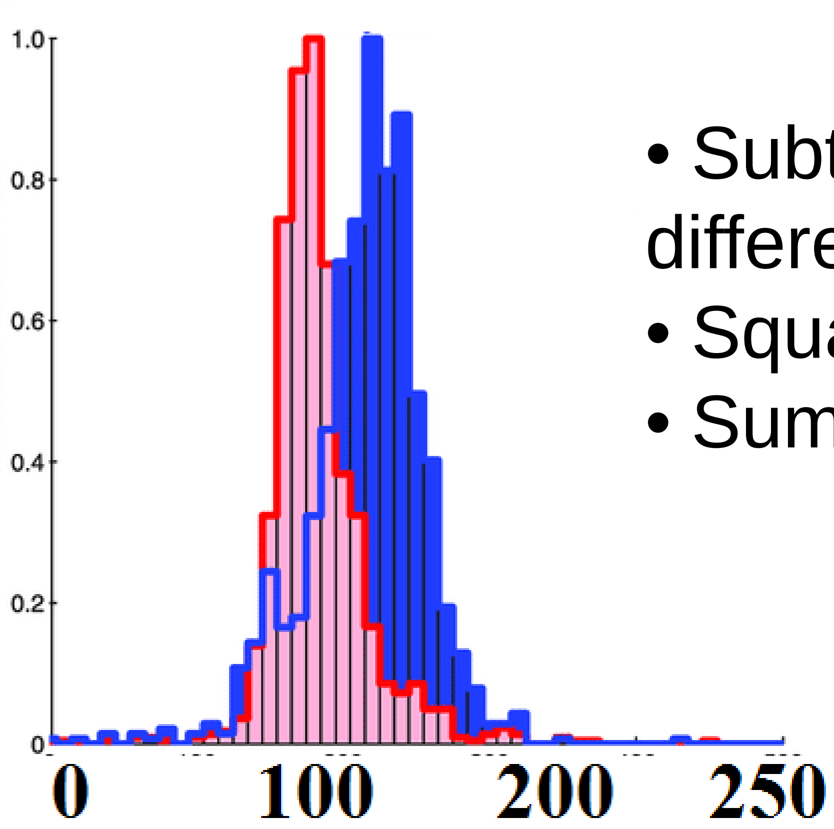
*Compare PDFs, P & Q using KLD*

*Store distance $d_{I\_Mk}$*

*end*

***<u>Choose the image which has lowest d</u>***

# Mean square distance




- Subtract the histograms to get difference
- Square the differences
- Sum the squared differences

# Algorithm

*Given model image set = {$M_1$, $M_2$, $M_3$..., $M_k$}*

*Given test image = I*

*for n=1:k*

*Read test image (I)*

*Read model image $M_k$*

*Generate two matrices corresponding to image I & $M_k$*

*Generate corresponding histograms ($h_1$ & $h_2$)*

*Normalize the histograms ($H_1$ & $H_2$)*

*They become PDF (P and Q)*

*Compute mean square distance*

*Store distance $d_{I\_Mk}$*

*end*

***Choose the image which has lowest d***