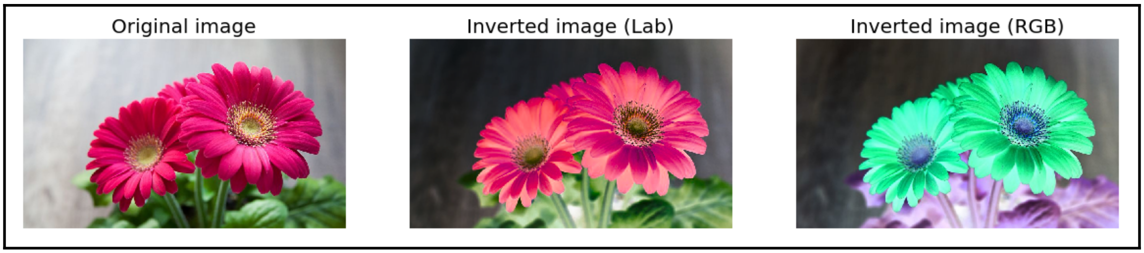
Chapter 1: Image Manipulation and Transformation



Identity

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Reflection

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scale

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Rotation

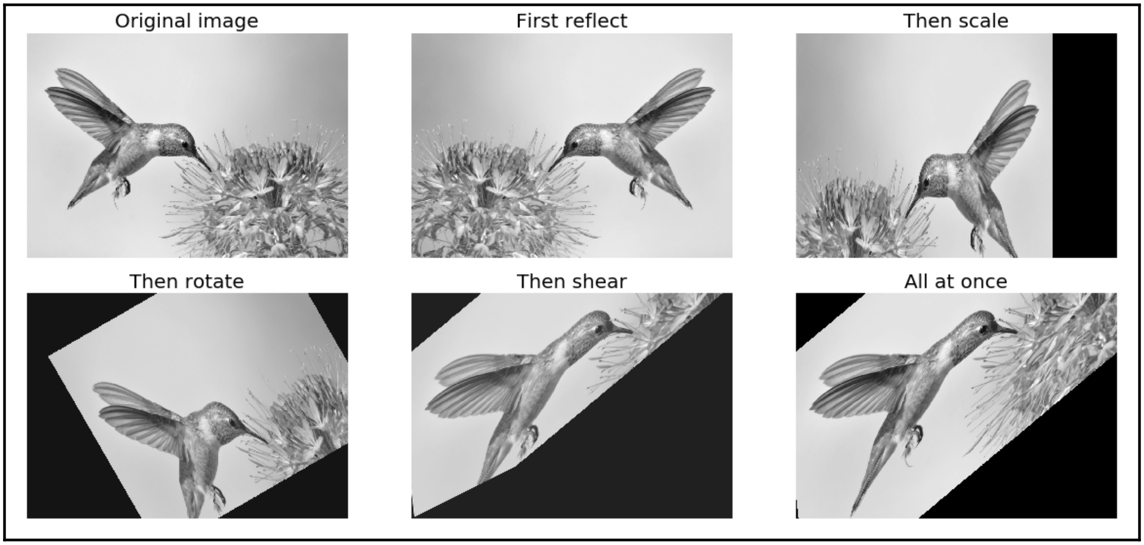
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

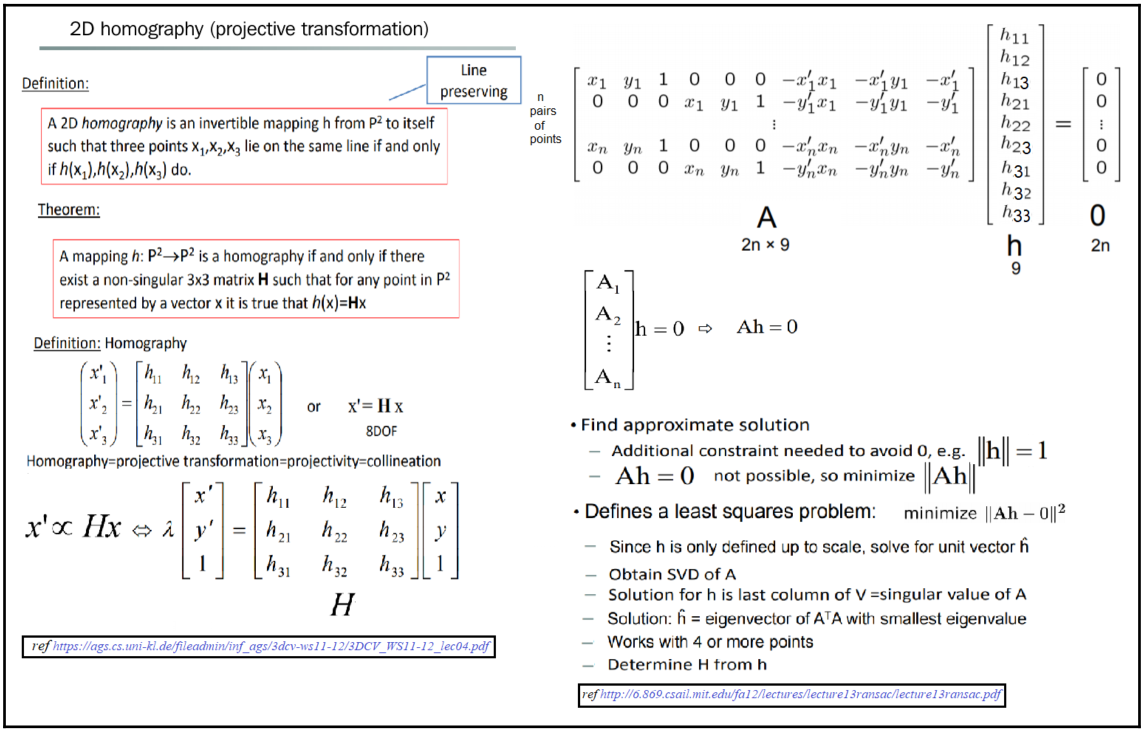
Shear-X

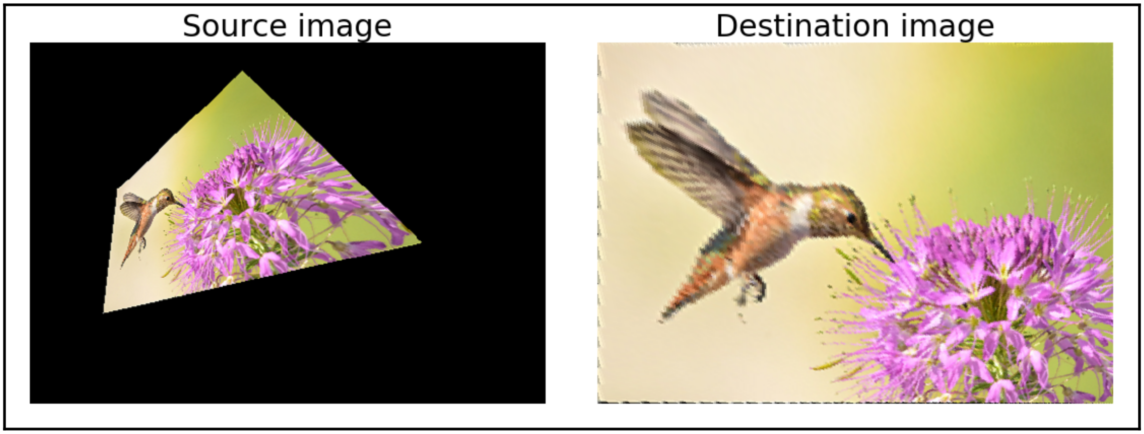
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \lambda_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

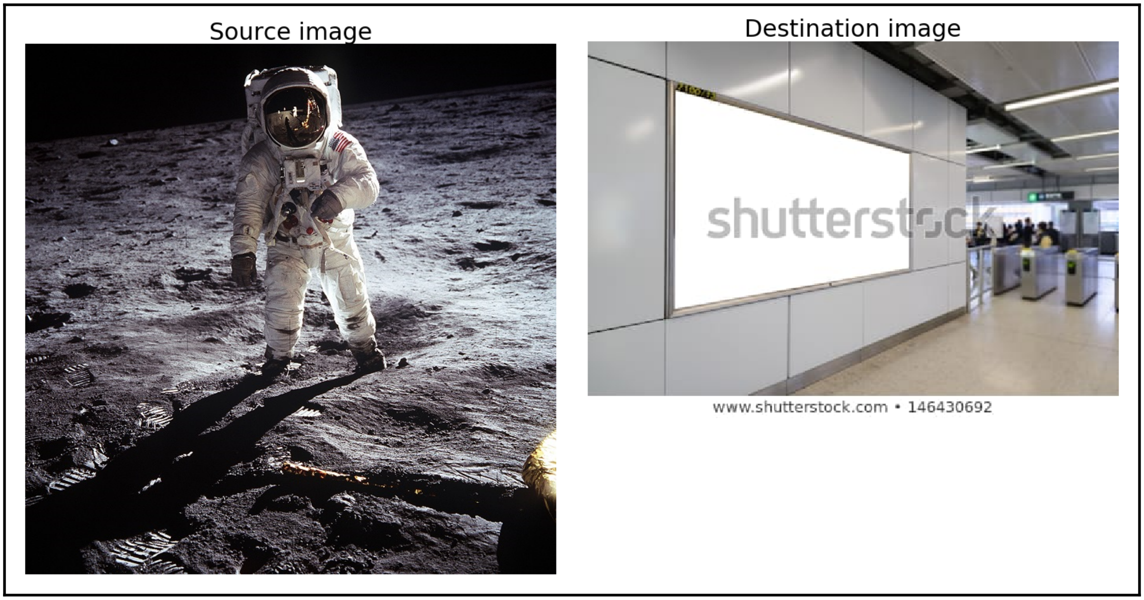
Shear-Y

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \lambda_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

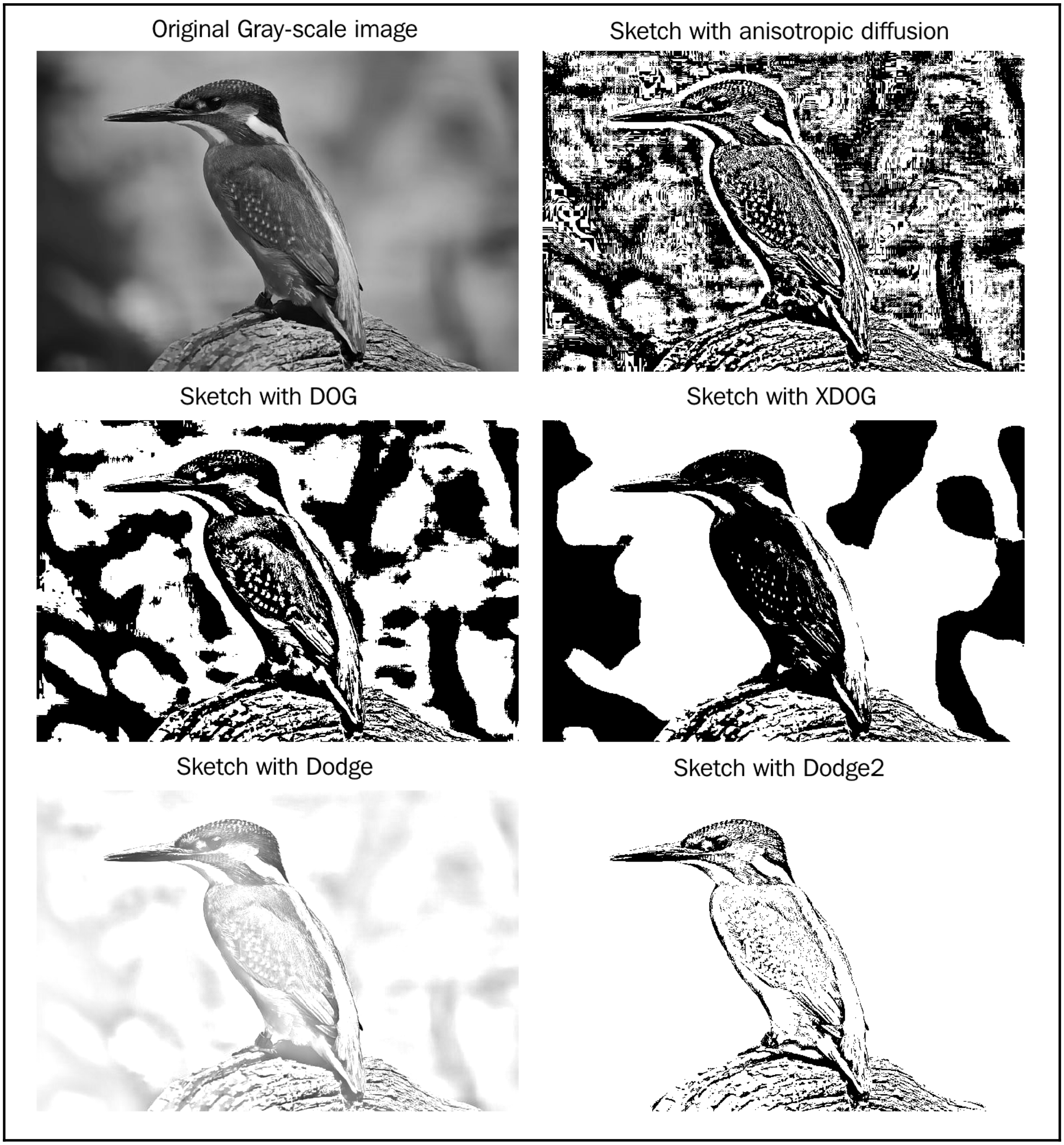

Original image
First reflect
Then scale
Then rotate
Then shear
All at once

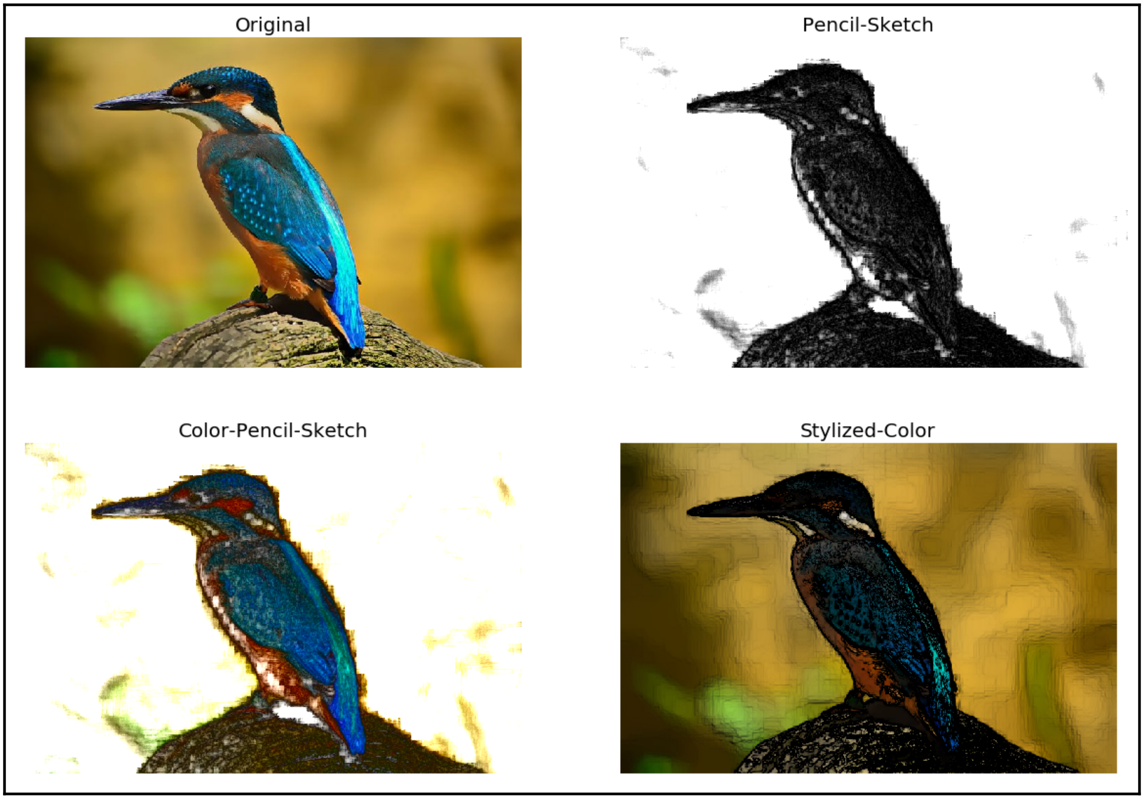

2D homography (projective transformation)
Definition:
Line preserving
A 2D homography is an invertible mapping h from P^2 to itself such that three points x_1, x_2, x_3 lie on the same line if and only if $h(x_1), h(x_2), h(x_3)$ do.
Theorem:
A mapping $h: P^2 \rightarrow P^2$ is a homography if and only if there exist a non-singular 3x3 matrix **H** such that for any point in P^2 represented by a vector x it is true that $h(x)=\mathbf{H}x$
Definition: Homography
$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ or $x' = \mathbf{H}x$
8DOF
Homography=projective transformation=projectivity=collineation
$x' \propto Hx \Leftrightarrow \lambda \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$
H
ref https://ags.cs.uni-kl.de/fileadmin/inf_ags/3dcv-ws11-12/3DCV_WS11-12_lec04.pdf
n pairs of points
$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 & -y'_1 \\ & & & & \vdots & & & & \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_nx_n & -y'_ny_n & -y'_n \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$
A
2n × 9
h
9
0
2n
$\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} h = 0 \Rightarrow Ah = 0$
• Find approximate solution
– Additional constraint needed to avoid 0, e.g. $\|h\| = 1$
– $Ah = 0$ not possible, so minimize $\|Ah\|$
• Defines a least squares problem: minimize $\|Ah - 0\|^2$
– Since h is only defined up to scale, solve for unit vector ĥ
– Obtain SVD of A
– Solution for h is last column of V =singular value of A
– Solution: ĥ = eigenvector of A^TA with smallest eigenvalue
– Works with 4 or more points
– Determine H from h
ref http://6.869.csail.mit.edu/fa12/lectures/lecture13ransac/lecture13ransac.pdf


Source image
Destination image


Source image
Destination image
shutterstock
www.shutterstock.com • 146430692


Output Image
www.shutterstock.com • 146430692

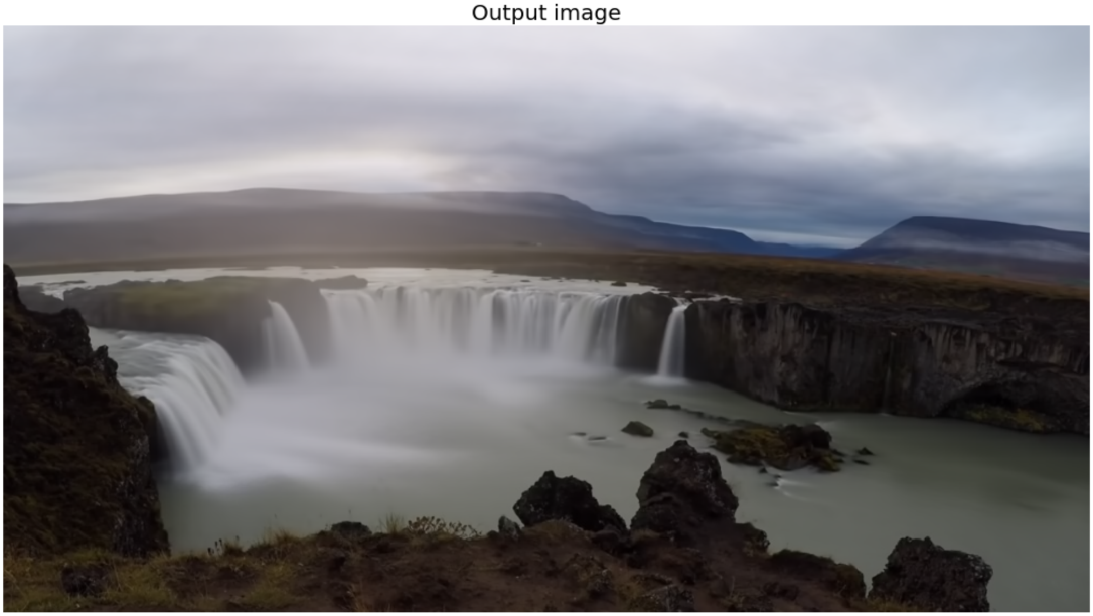

Original Gray-scale image
Sketch with anisotropic diffusion
Sketch with DOG
Sketch with XDOG
Sketch with Dodge
Sketch with Dodge2


Original
Pencil-Sketch
Color-Pencil-Sketch
Stylized-Color


Original Image
Cartoonized Image

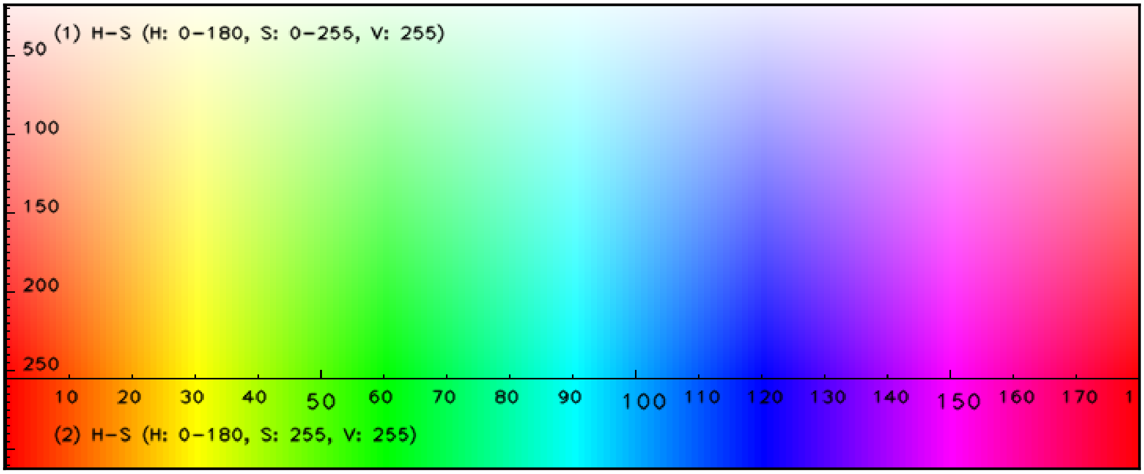
An input image

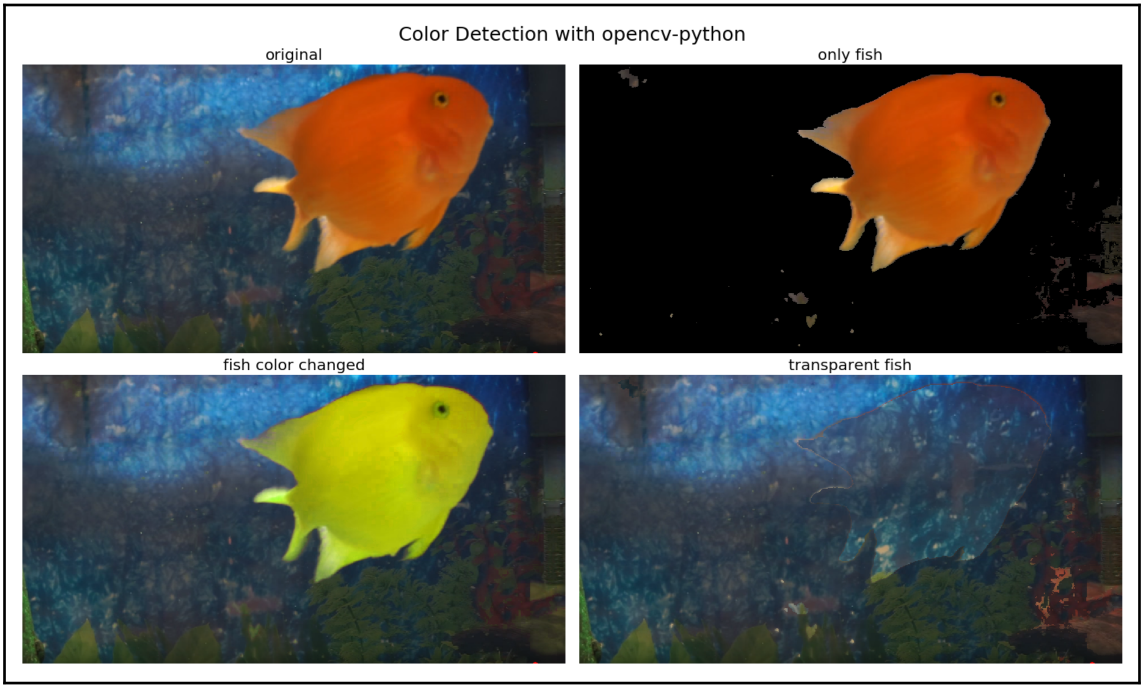
Output image



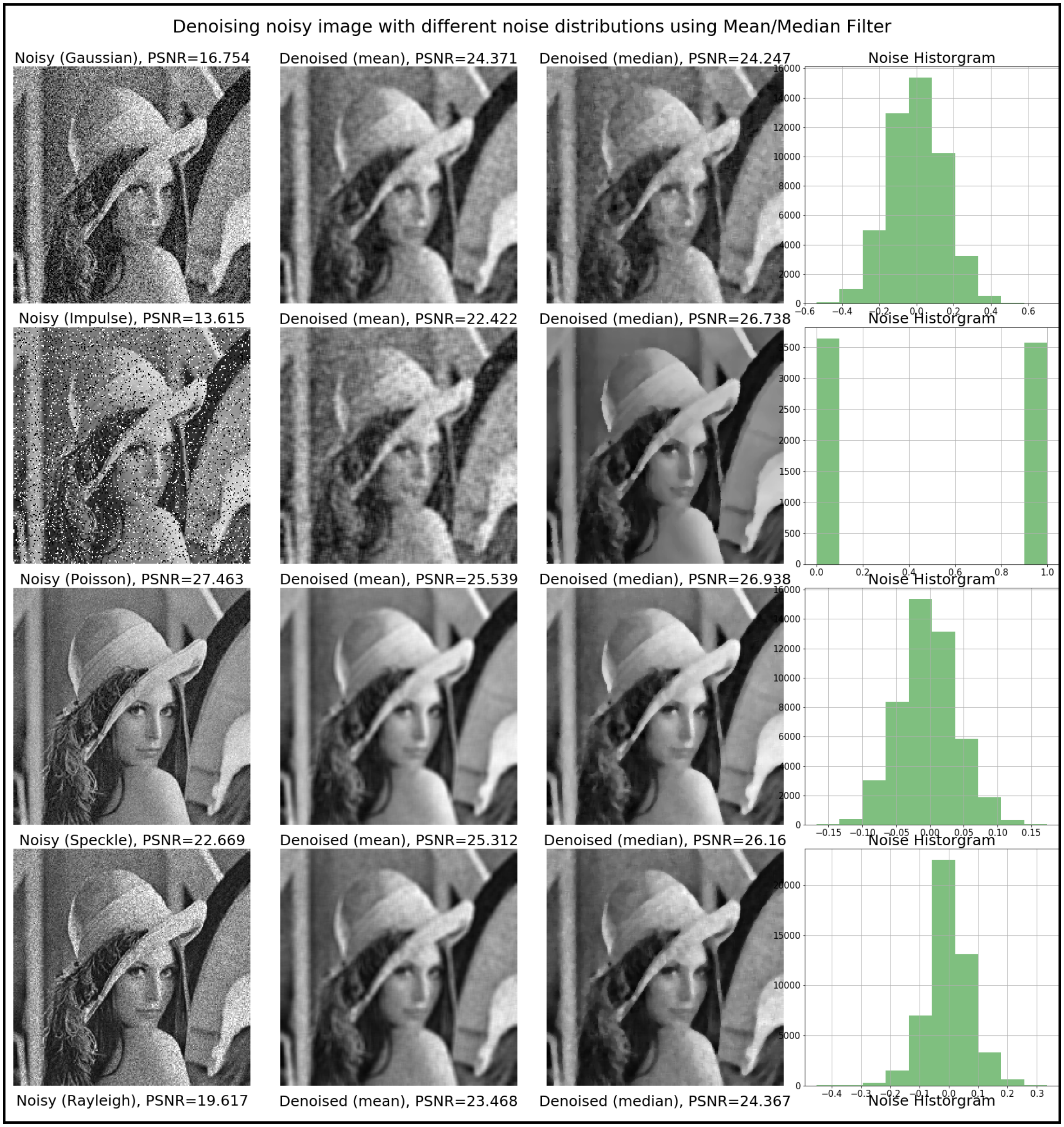
An input image

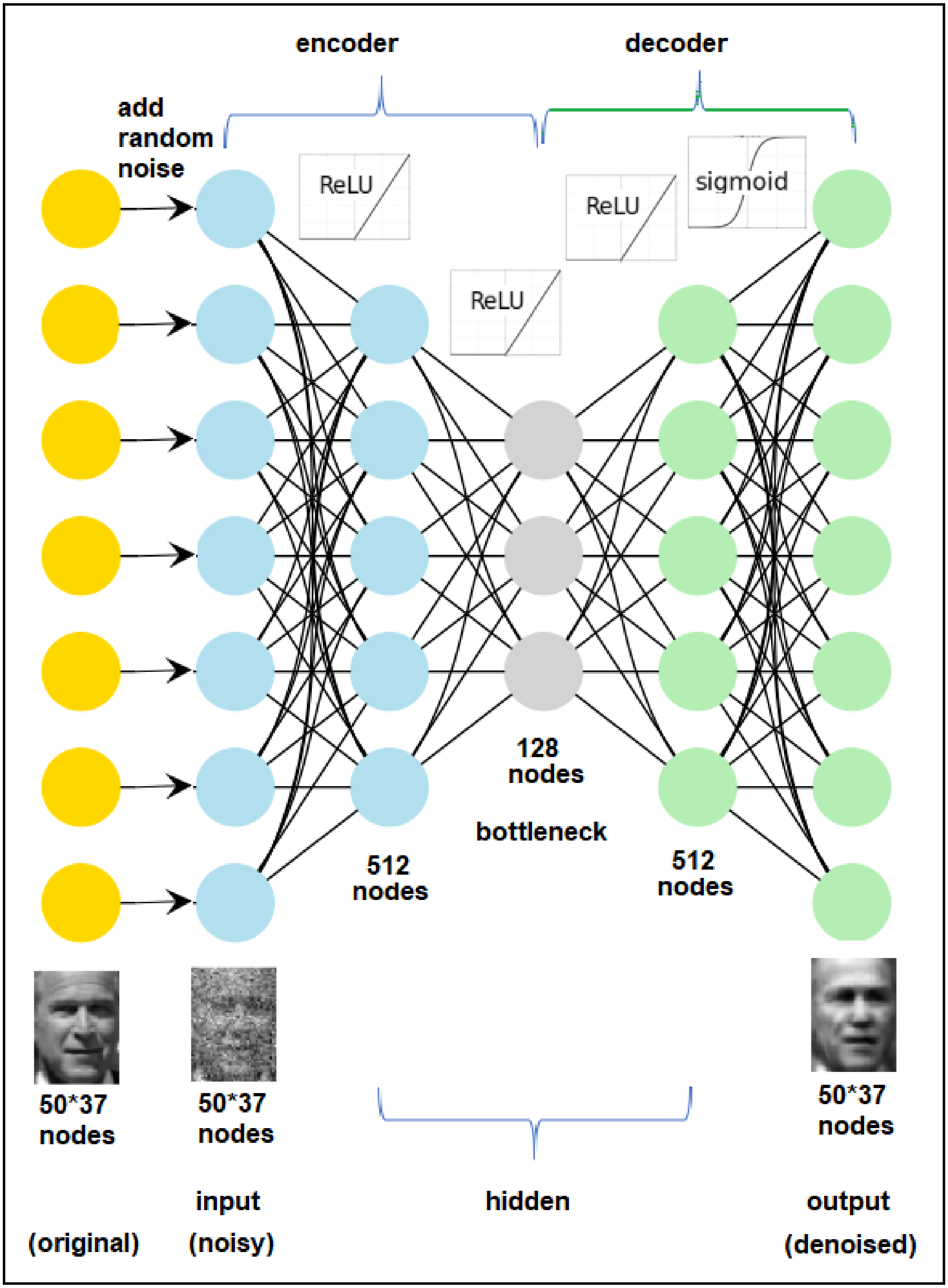
Output image

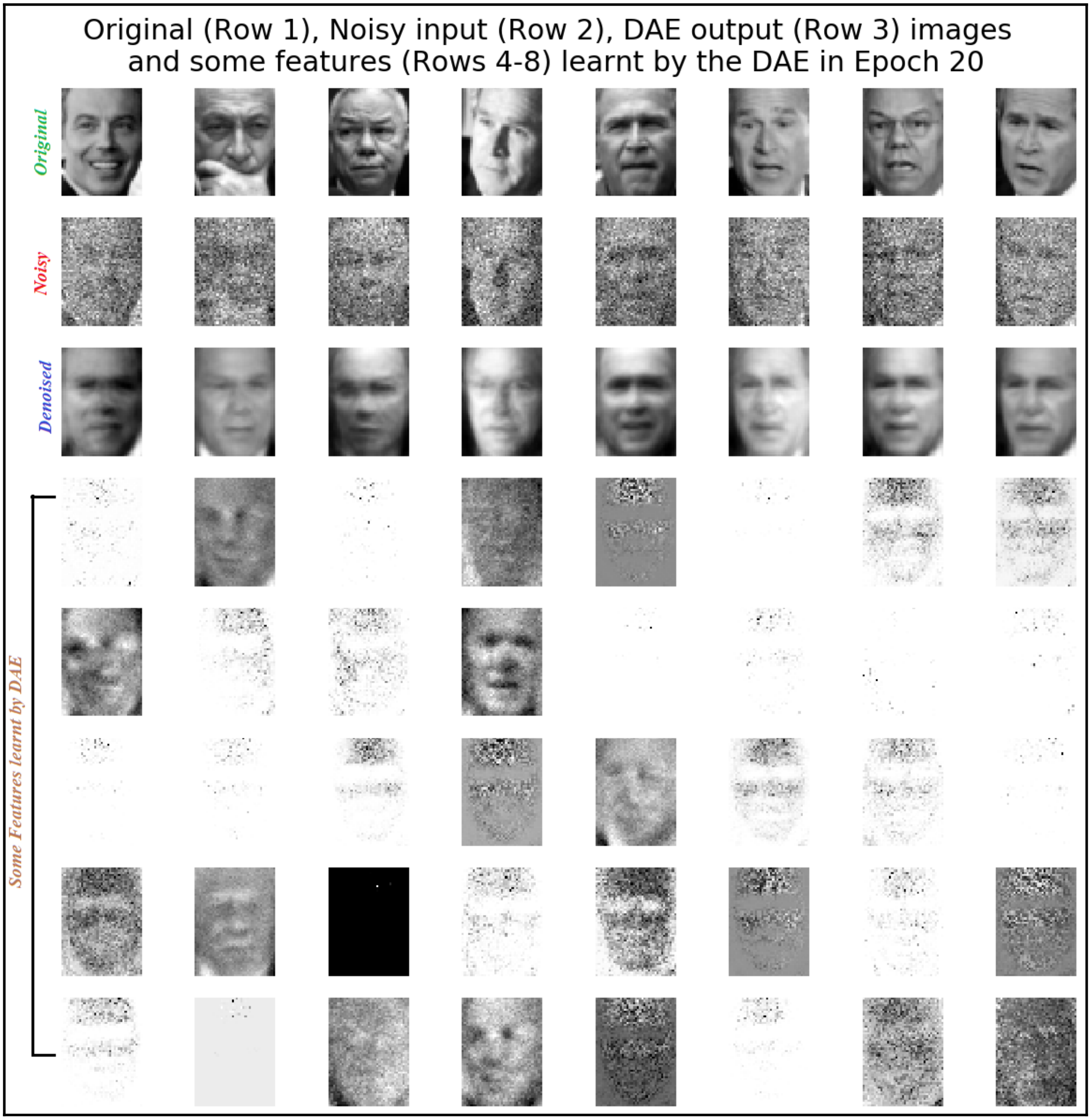

(1) H-S (H: 0-180, S: 0-255, V: 255)
50
100
150
200
250
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
(2) H-S (H: 0-180, S: 255, V: 255)

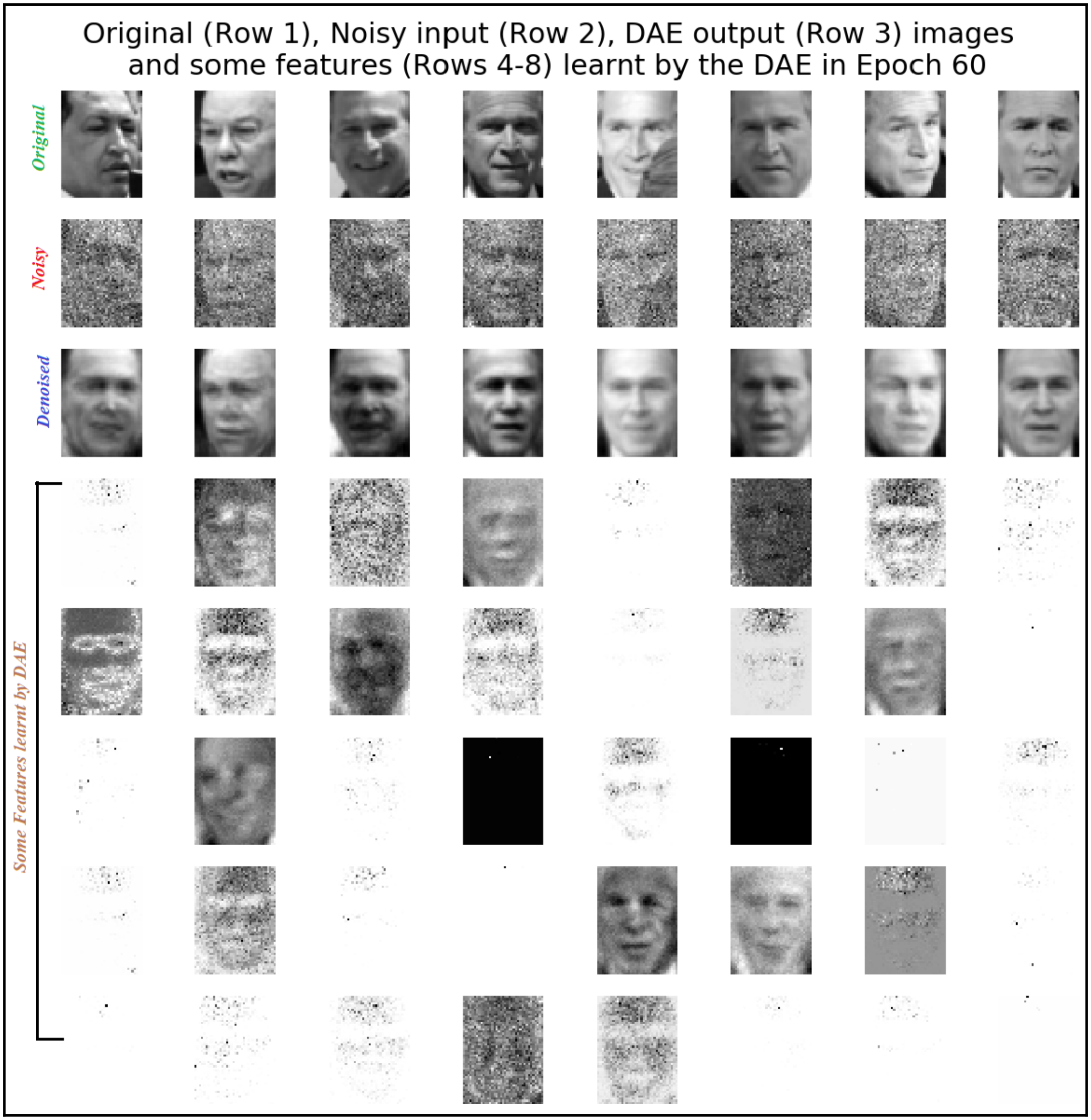

Color Detection with opencv-python
original
only fish
fish color changed
transparent fish

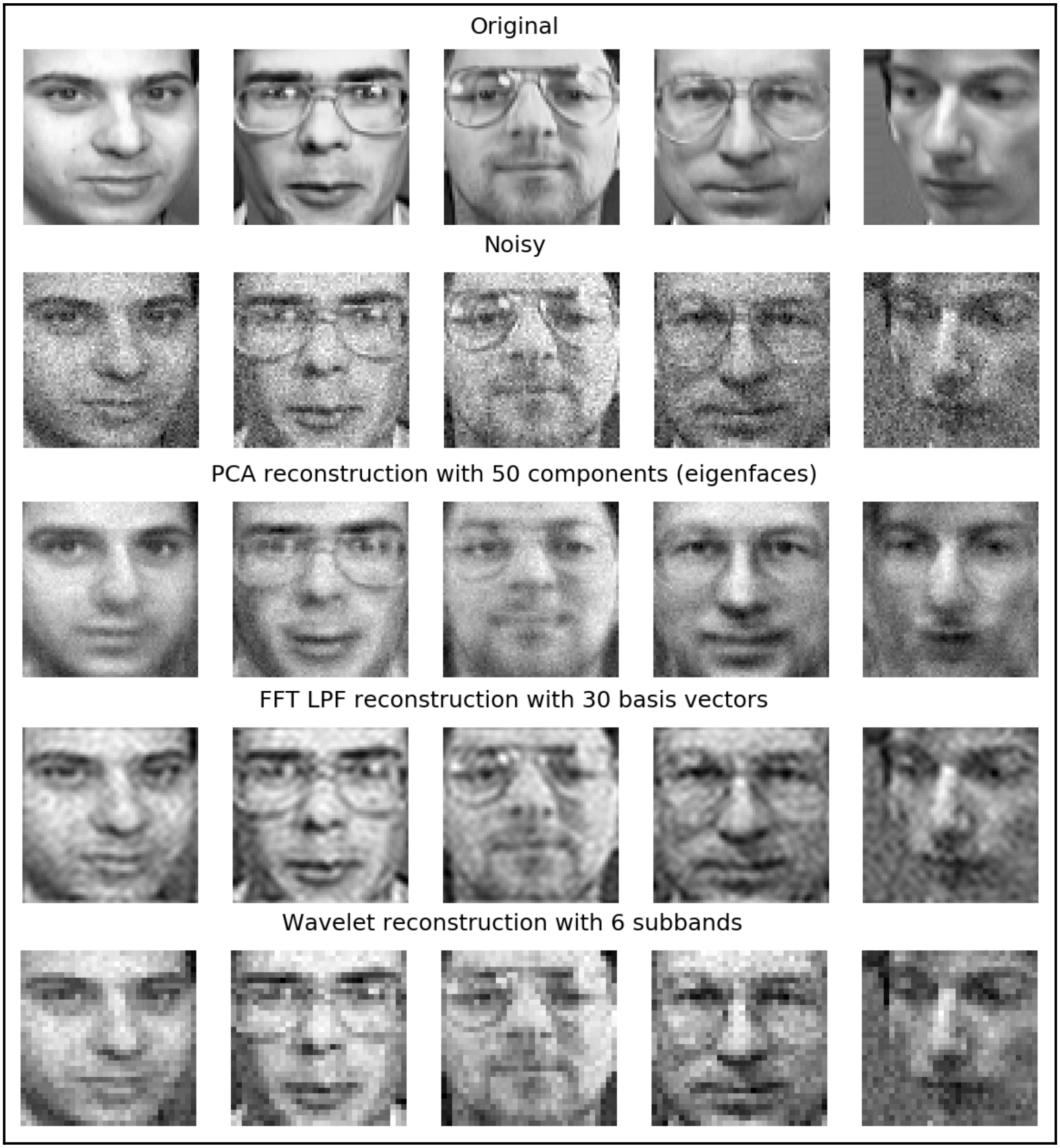
Chapter 2: Image Enhancement




encoder
decoder
add
random
noise
ReLU
ReLU
ReLU
sigmoid
128
nodes
bottleneck
512
nodes
512
nodes
50*37
nodes
50*37
nodes
50*37
nodes
input
hidden
output
(original)
(noisy)
(denoised)


Original (Row 1), Noisy input (Row 2), DAE output (Row 3) images and some features (Rows 4-8) learnt by the DAE in Epoch 20
Original
Noisy
Denoised
Some Features learnt by DAE


Original (Row 1), Noisy input (Row 2), DAE output (Row 3) images
and some features (Rows 4-8) learnt by the DAE in Epoch 60
Original
Noisy
Denoised
Some Features learnt by DAE


Original
Noisy
PCA reconstruction with 50 components (eigenfaces)
FFT LPF reconstruction with 30 basis vectors
Wavelet reconstruction with 6 subbands


Original
Noisy
Anisotropic Diffusion (Perona Malik eq 1, iter=20, $\kappa = 20$)
Anisotropic Diffusion (Perona Malik eq 2, iter=50, $\kappa = 50$)

Anisotropic Heat Diffusion Equation

$$I_t = div\,(c(x,y,t)\nabla I) = c(x,y,t).\,\Delta I + \nabla c \cdot \nabla I$$

Image at time t, divergence, Gradient, conductivity function, Laplacian

$$= \nabla \cdot (c\,\nabla I) = \frac{\partial}{\partial x}(c\,I_x) + \frac{\partial}{\partial y}(c\,I_y)$$

calculate the conductivity function c every iteration based on the current image I

$$c(x,y,t) = g(\|\nabla I(x,y,t)\|)$$

Perona-Malik edge-stopping function

$$g(\nabla I) = e^{(-(\|\nabla I\|/K)^2)} \qquad \text{(equation 1)}$$

$$g(\nabla I) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2} \qquad \text{(equation 2)}$$

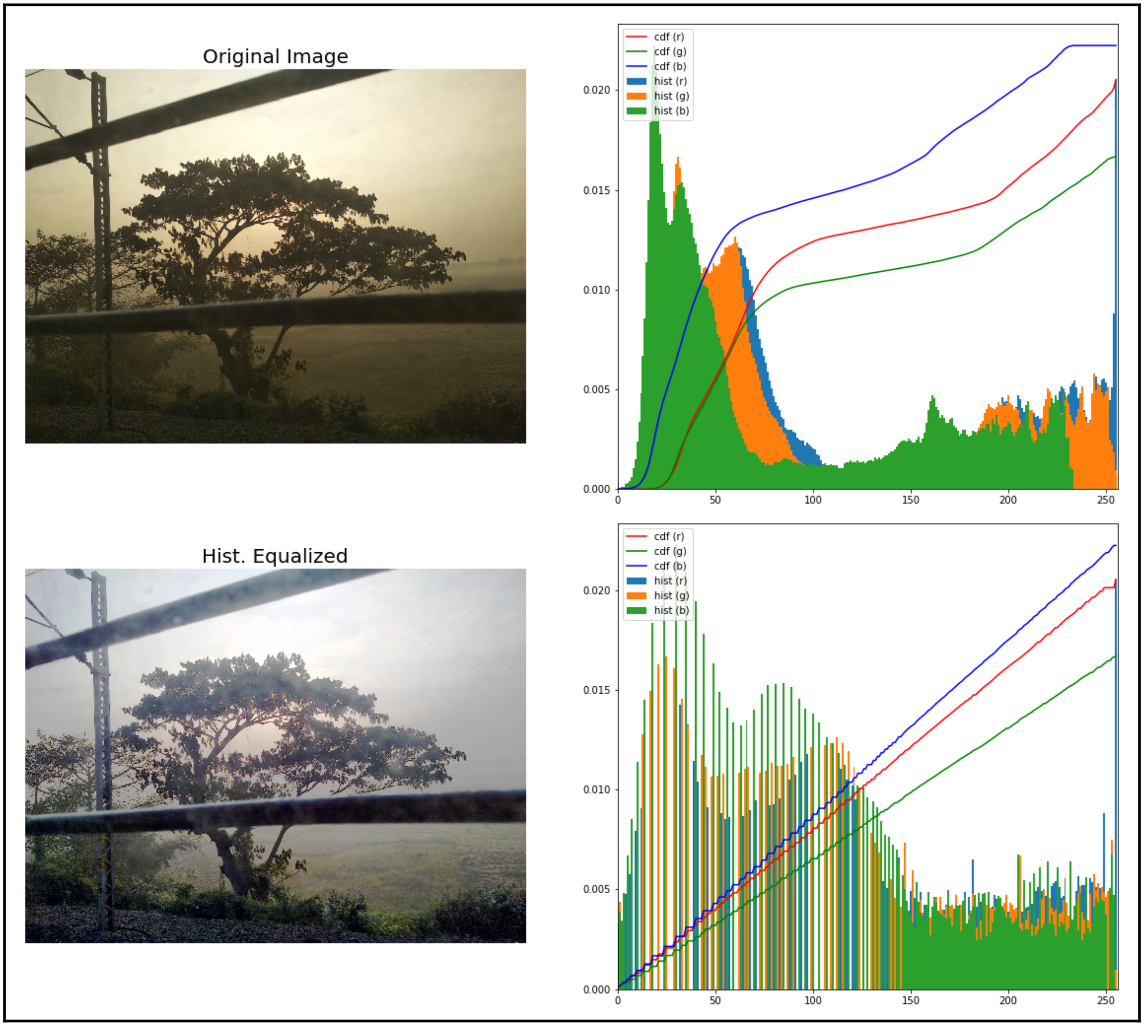
Taken from http://image.diku.dk/imagecanon/material/PeronaMalik1990.pdf

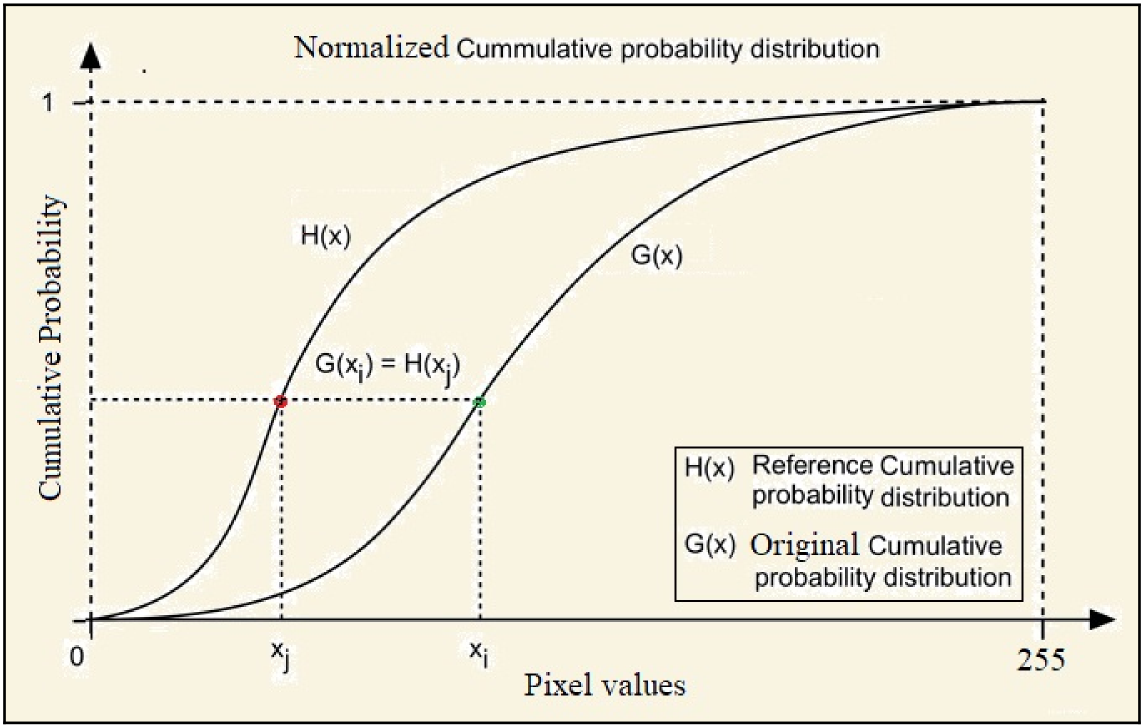
$$s_k = T(r_k) = \sum_{j=0}^{k} P_r(r_j) = \sum_{j=0}^{k} n_j / N$$

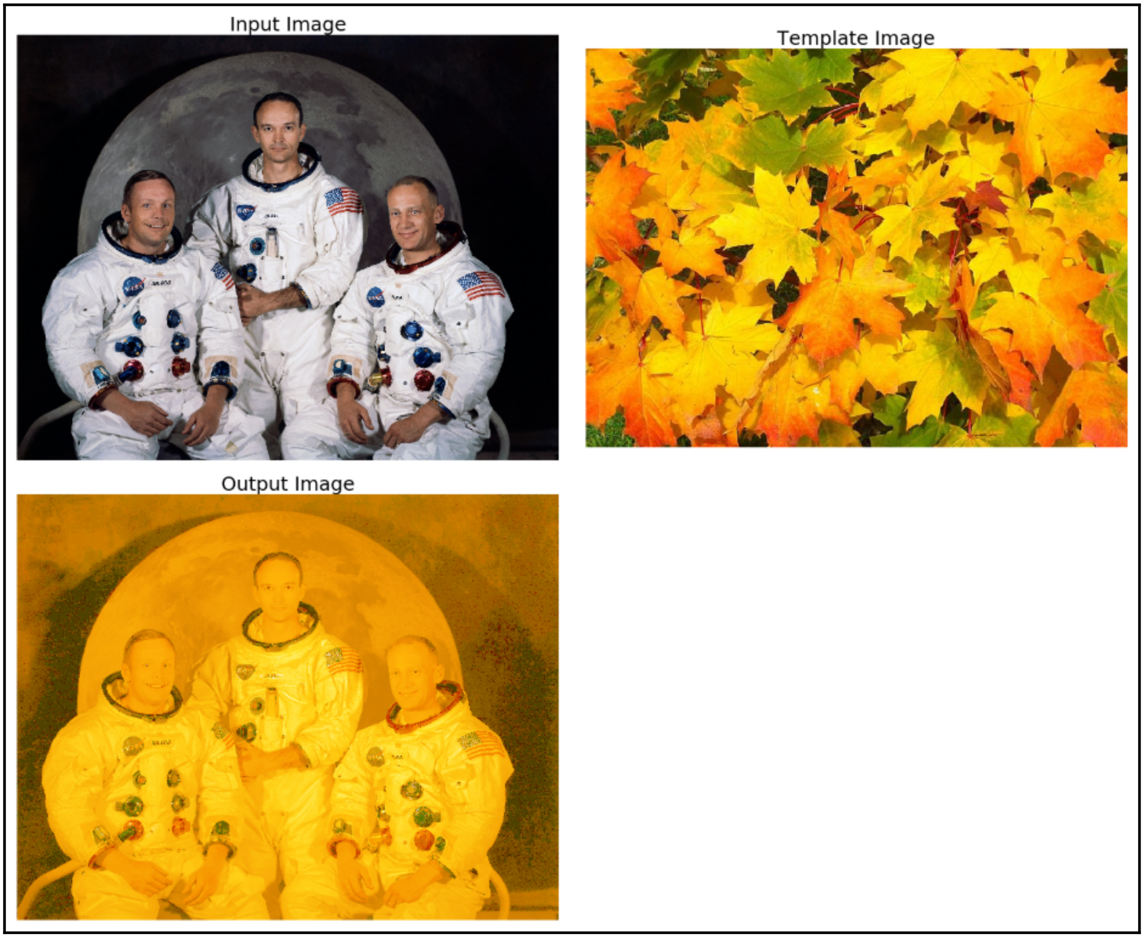
$$0 \le r_k \le 1, \qquad k = 0,1,2,\ldots\ldots,255$$

N: total number of pixels

n_j: frequency of pixel with gray-level j

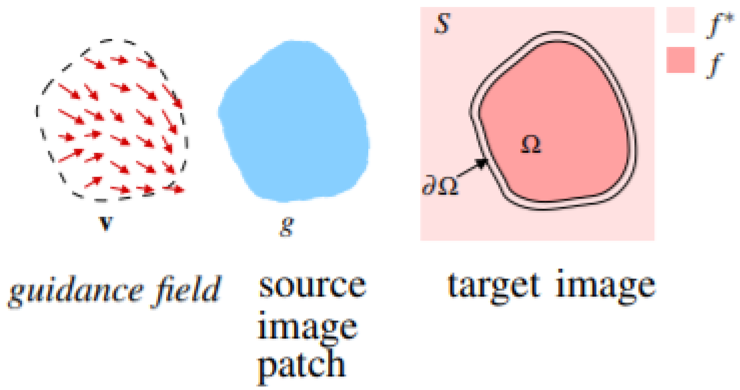

Original Image
Hist. Equalized
cdf (r)
cdf (g)
cdf (b)
hist (r)
hist (g)
hist (b)
0.000
0.005
0.010
0.015
0.020
0
50
100
150
200
250


Normalized Cummulative probability distribution
1
Cumulative Probability
H(x)
G(x)
G(x_i) = H(x_j)
H(x) Reference Cumulative probability distribution
G(x) Original Cumulative probability distribution
0
x_j
x_i
255
Pixel values


Input Image
Template Image
Output Image


Input Image
Template Image
Output Image

Guided Interpolation



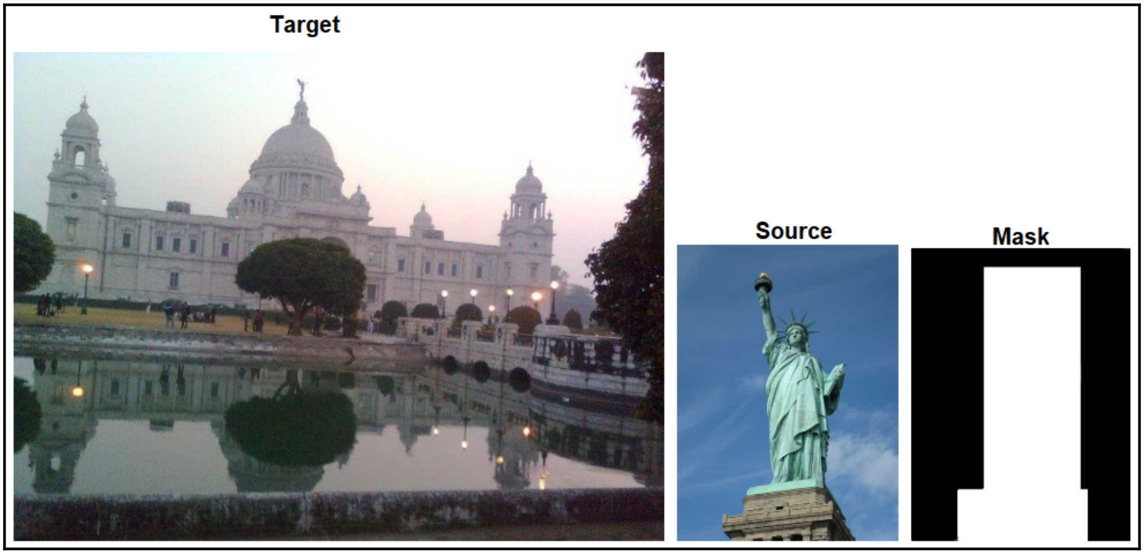
$\boxed{\text{membrane interpolant } f \text{ of } f^* \text{ over } \Omega}$

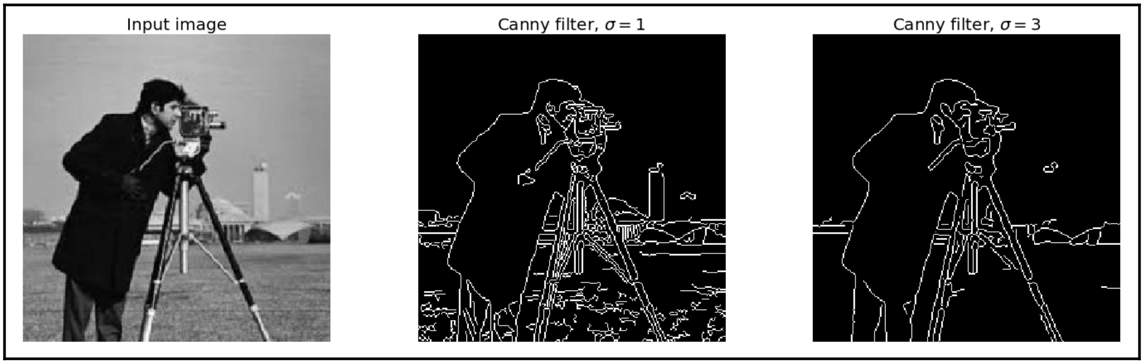
Solve $\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$

By Euler-Lagrange equation

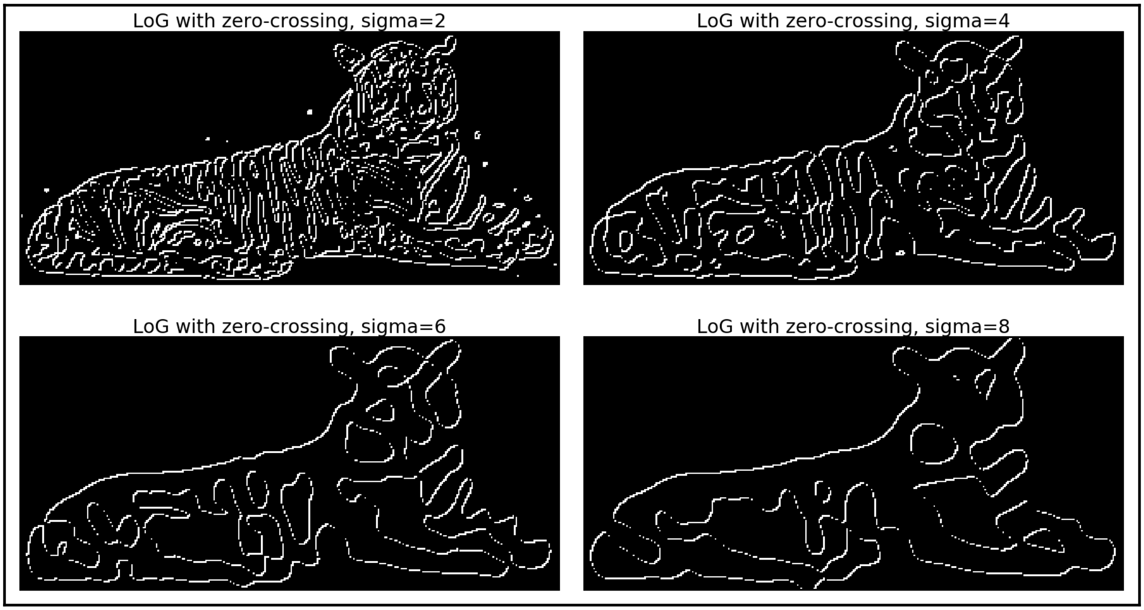
$$\underbrace{\Delta f = \nabla \cdot \mathbf{v} \text{ over } \Omega}_{\text{Poisson equation}}, \underbrace{\text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega}}_{\text{Dirichlet boundary conditions}},$$

Taken from
https://www.cs.virginia.edu/~connelly/class/2014/comp_photo/proj2/poisson.pdf

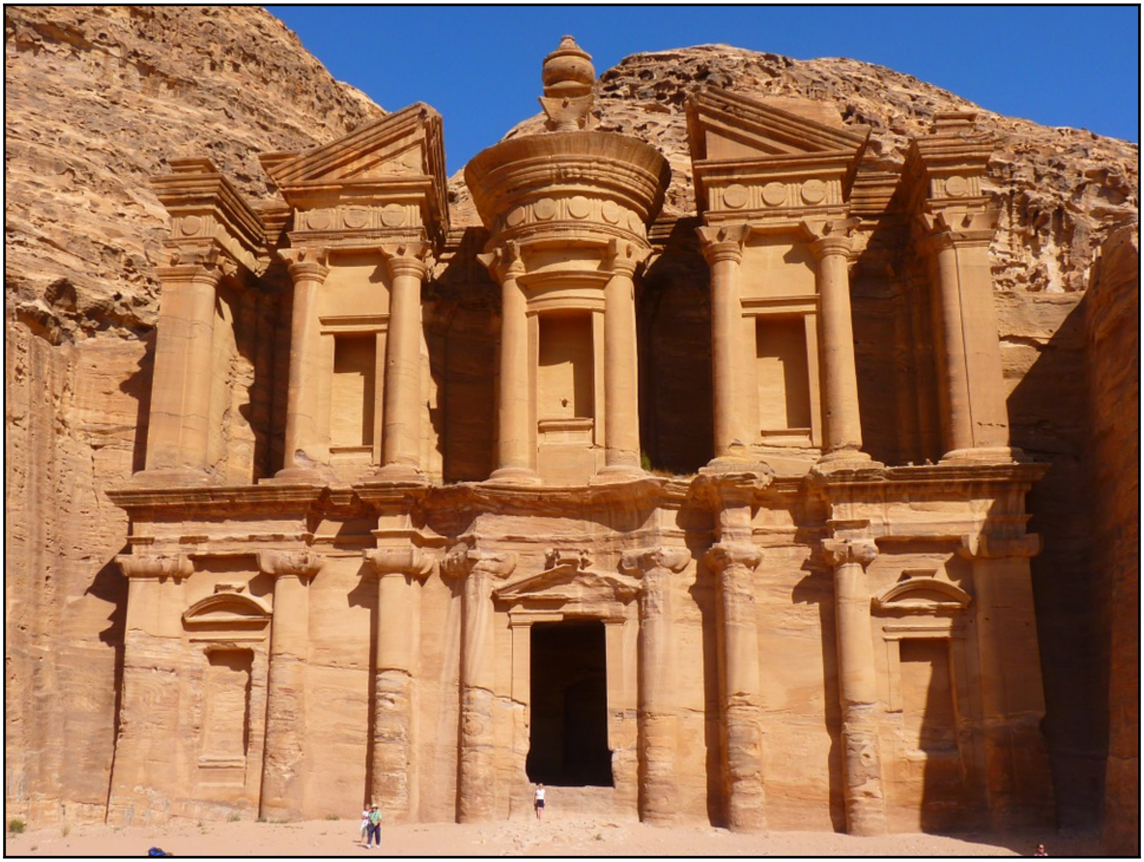

Target
Source
Mask

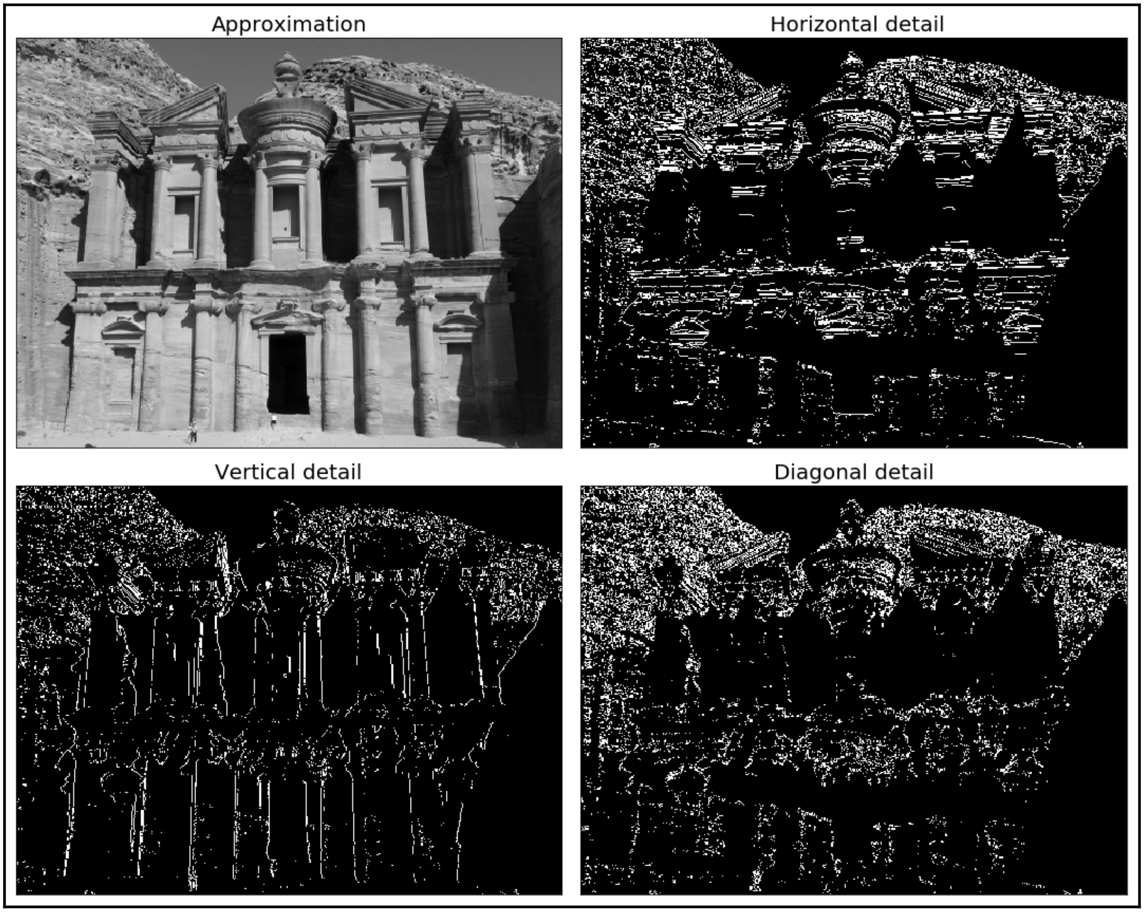


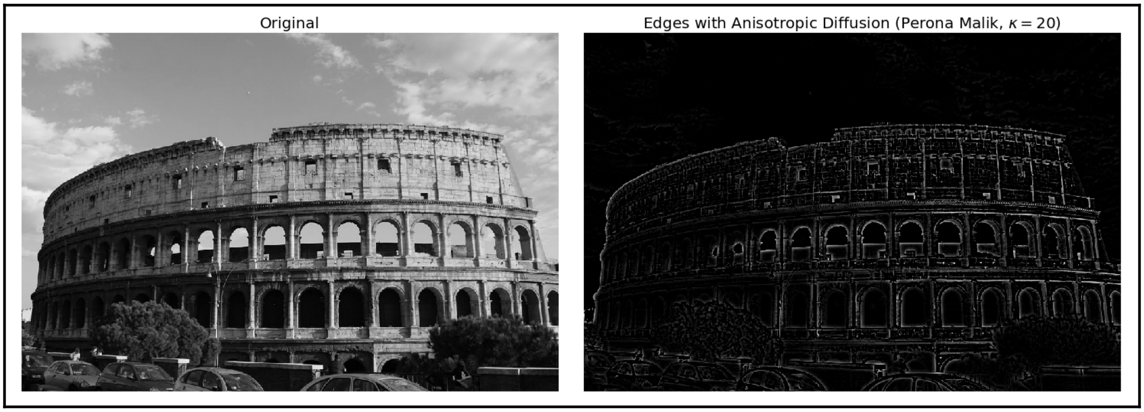
Input image
Canny filter, $\sigma = 1$
Canny filter, $\sigma = 3$



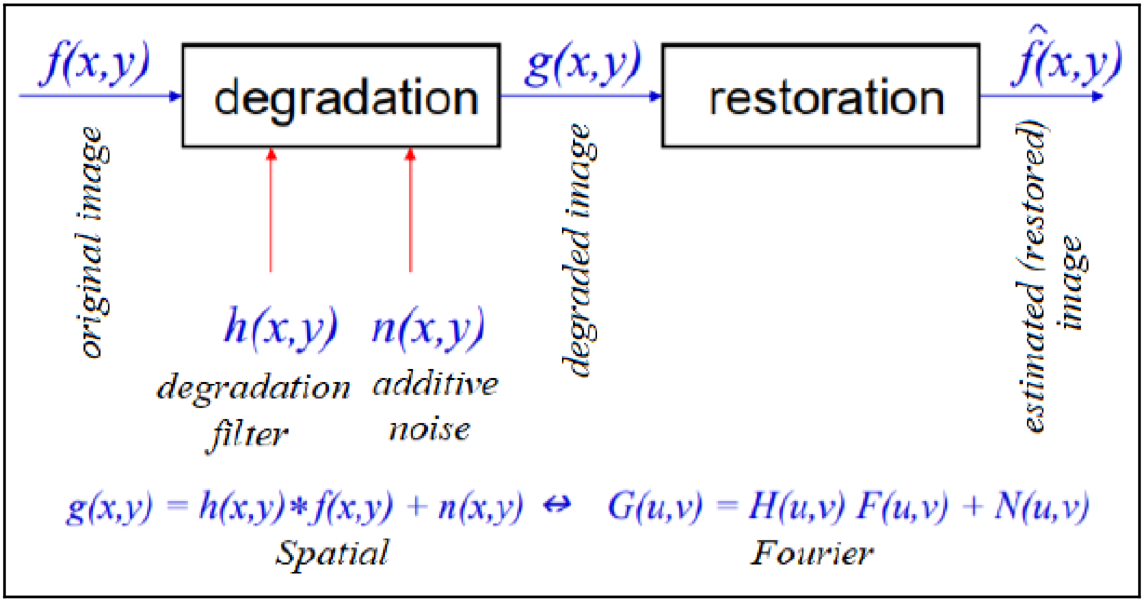
LoG with zero-crossing, sigma=2
LoG with zero-crossing, sigma=4
LoG with zero-crossing, sigma=6
LoG with zero-crossing, sigma=8



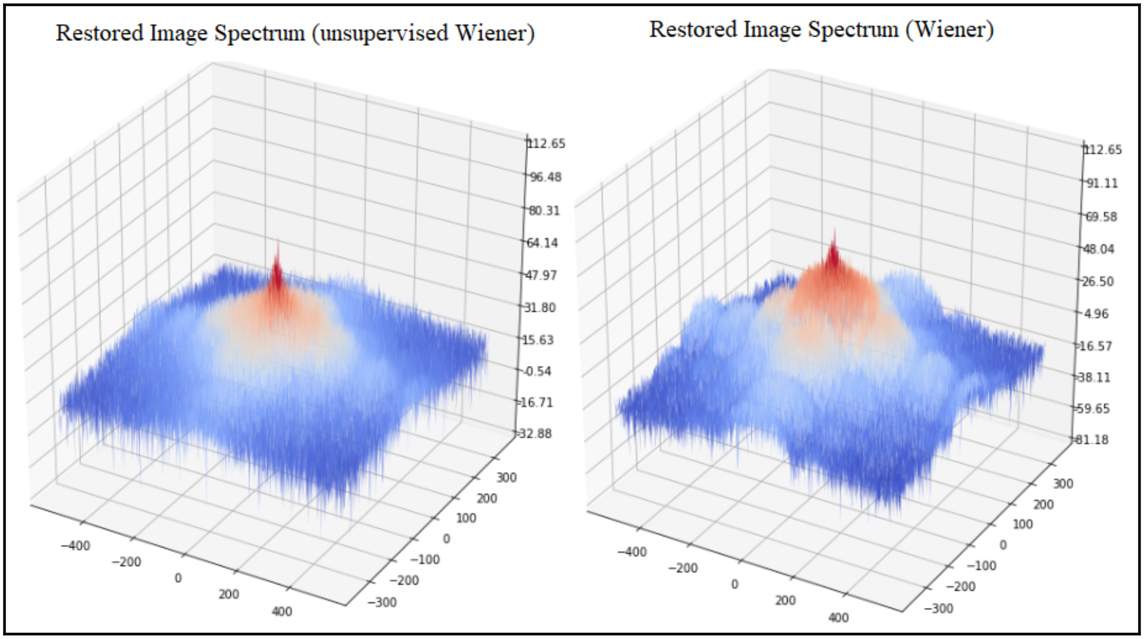

Approximation
Horizontal detail
Vertical detail
Diagonal detail


Original
Edges with Anisotropic Diffusion (Perona Malik, κ = 20)

Chapter 3: Image Restoration




Original image
Noisy blurred image: PSNR=24.099
Self tuned Wiener restoration: PSNR=24.664
Wiener restoration: PSNR=26.036


Restored Image Spectrum (unsupervised Wiener)
Restored Image Spectrum (Wiener)

Wiener Filter

objective $\min_W E[(f - \hat{f})^2]$

$$= \min_W E[\,|F(u,v) - \hat{F}(u,v)|^2]$$ by Parseval's Theorem

s.t. $\hat{F}(u,v) = G(u,v)W(u,v)$

solution:

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \frac{|N(u,v)|^2}{|F(u,v)|^2}} = \frac{H^*(u,v)}{|H(u,v)|^2 + K}$$

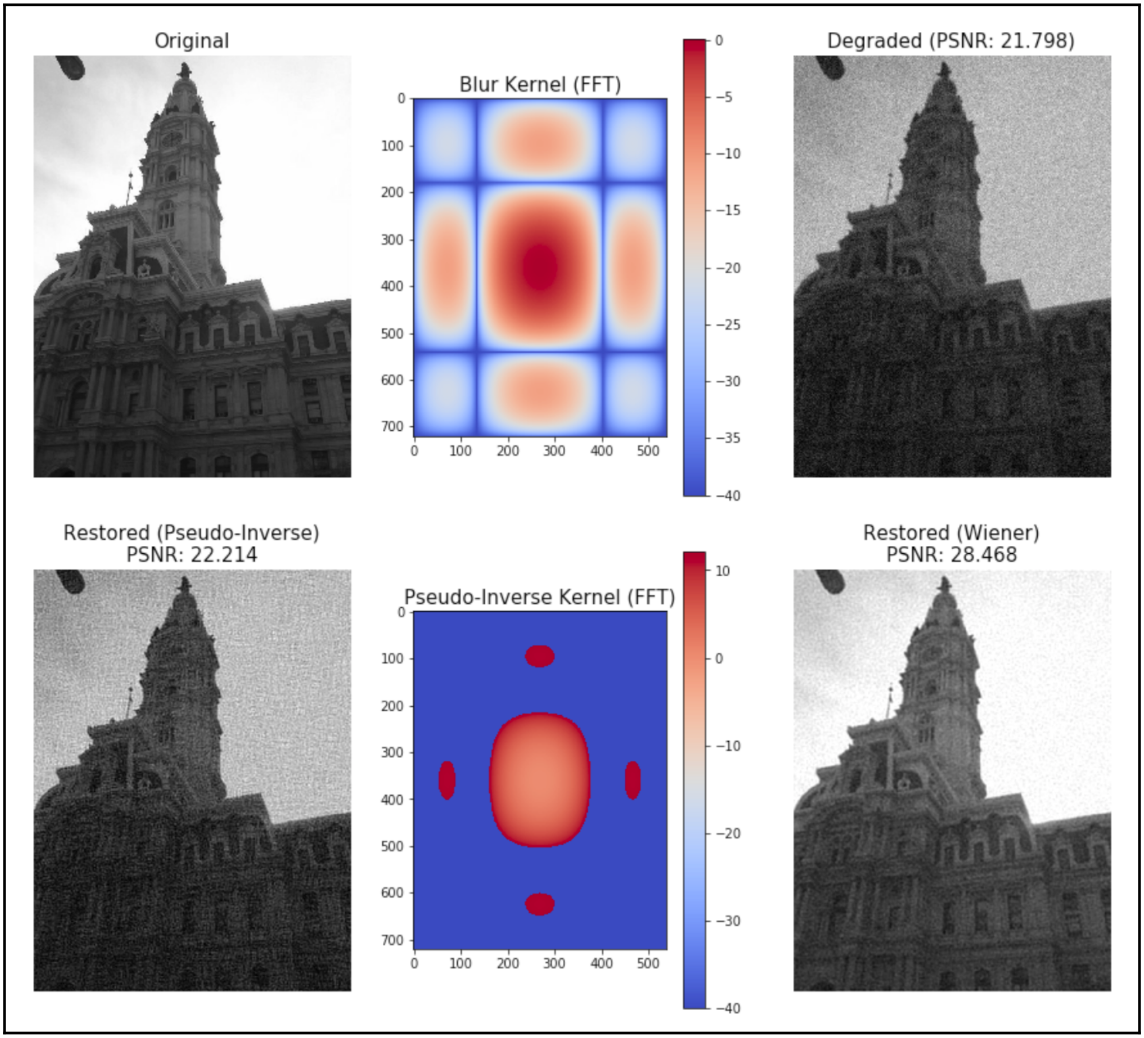
$$= \boxed{\frac{1}{H(u,v)}} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{|N(u,v)|^2}{|F(u,v)|^2}}$$

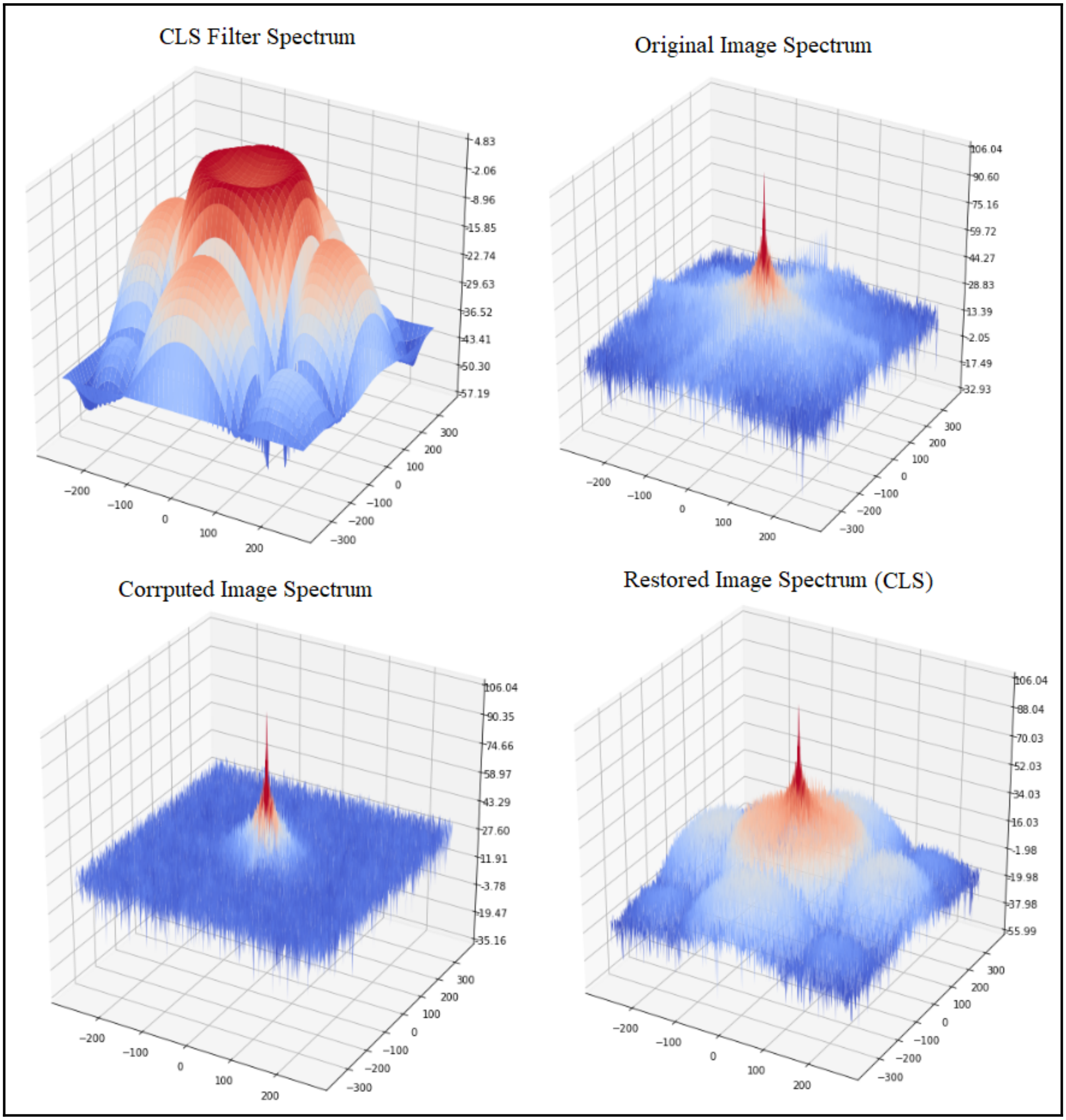
inverse filter 1/SNR

$$W(u,v) = \frac{H^*(u,v)}{|H(u,v)|^2 + \lambda|\Lambda_D|^2}$$

scikit-image restoration's wiener()

balance Freq. response of Laplacian


Original
Blur Kernel (FFT)
Degraded (PSNR: 21.798)
Restored (Pseudo-Inverse)
PSNR: 22.214
Pseudo-Inverse Kernel (FFT)
Restored (Wiener)
PSNR: 28.468


CLS Filter Spectrum
Original Image Spectrum
Corrputed Image Spectrum
Restored Image Spectrum (CLS)

Constrained Least-Squares Filter

$g = Hf + n$ (degradation equation)

objective: $\min_f \| g - Hf \|_2^2$

s.t. $\| Cf \|_2^2 < \varepsilon$ (smoothness constraint)

Data fidelity Smoothness

objective: $\min_f \left(\| g - Hf \|_2^2 + \lambda \| Cf \|_2^2 \right)$

Lagrange multiplier

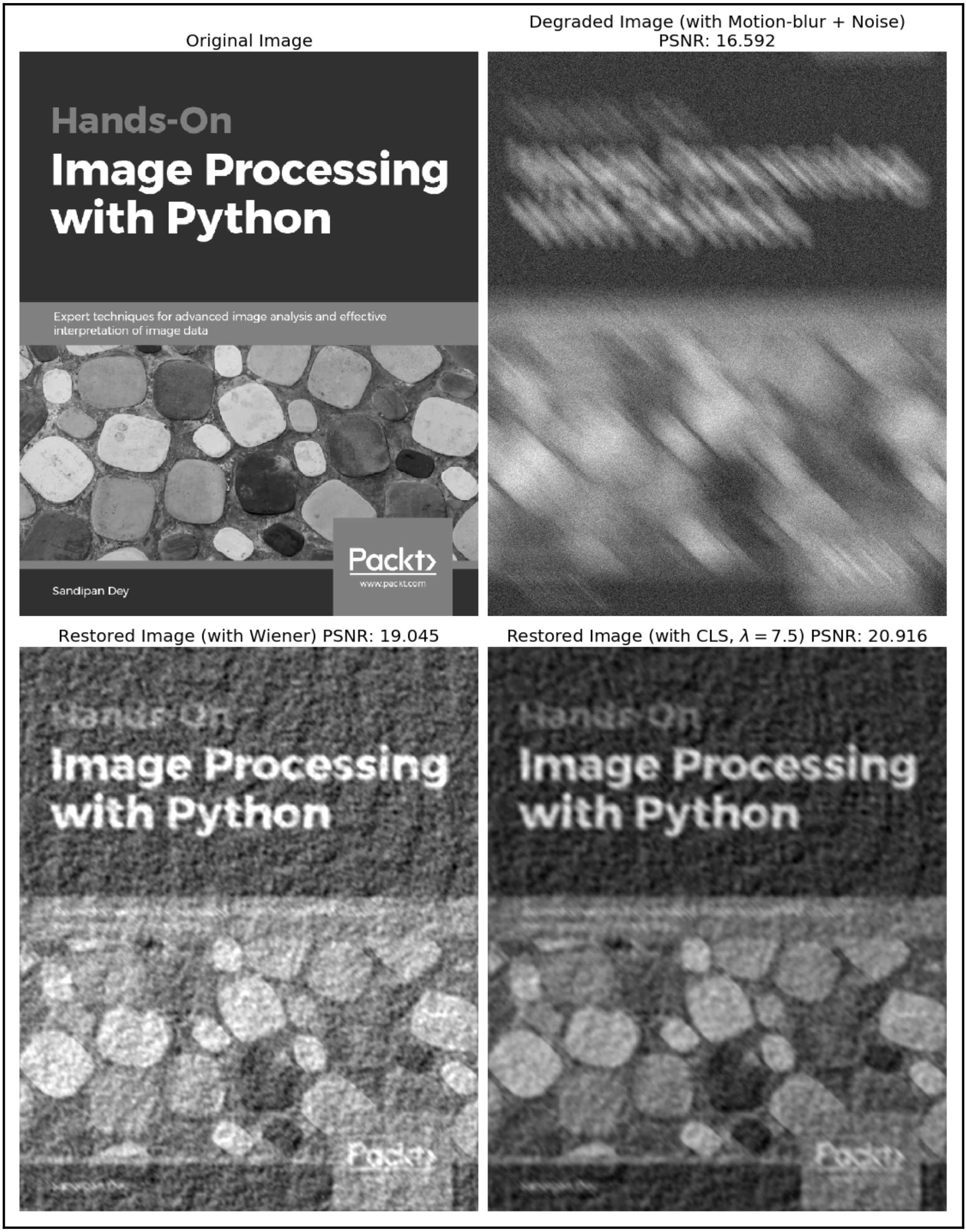
solution: $\hat{f} = (H^T H + \lambda C^T C)^{+} (H^T g)$

restored Lagrange multiplier (regularization parameter) high-pass filter

$$\hat{F}(u,v) = \frac{H^*(u,v)\, G(u,v)}{\|H(u,v)\|^2 + \lambda \|C(u,v)\|^2}$$

restored

(in freq. domain)


Original Image
Hands-On
Image Processing
with Python
Expert techniques for advanced image analysis and effective
interpretation of image data
Packt>
www.packt.com
Sandipan Dey
Degraded Image (with Motion-blur + Noise)
PSNR: 16.592
Restored Image (with Wiener) PSNR: 19.045
Image Processing
with Python
Restored Image (with CLS, λ = 7.5) PSNR: 20.916
Image Processing
with Python

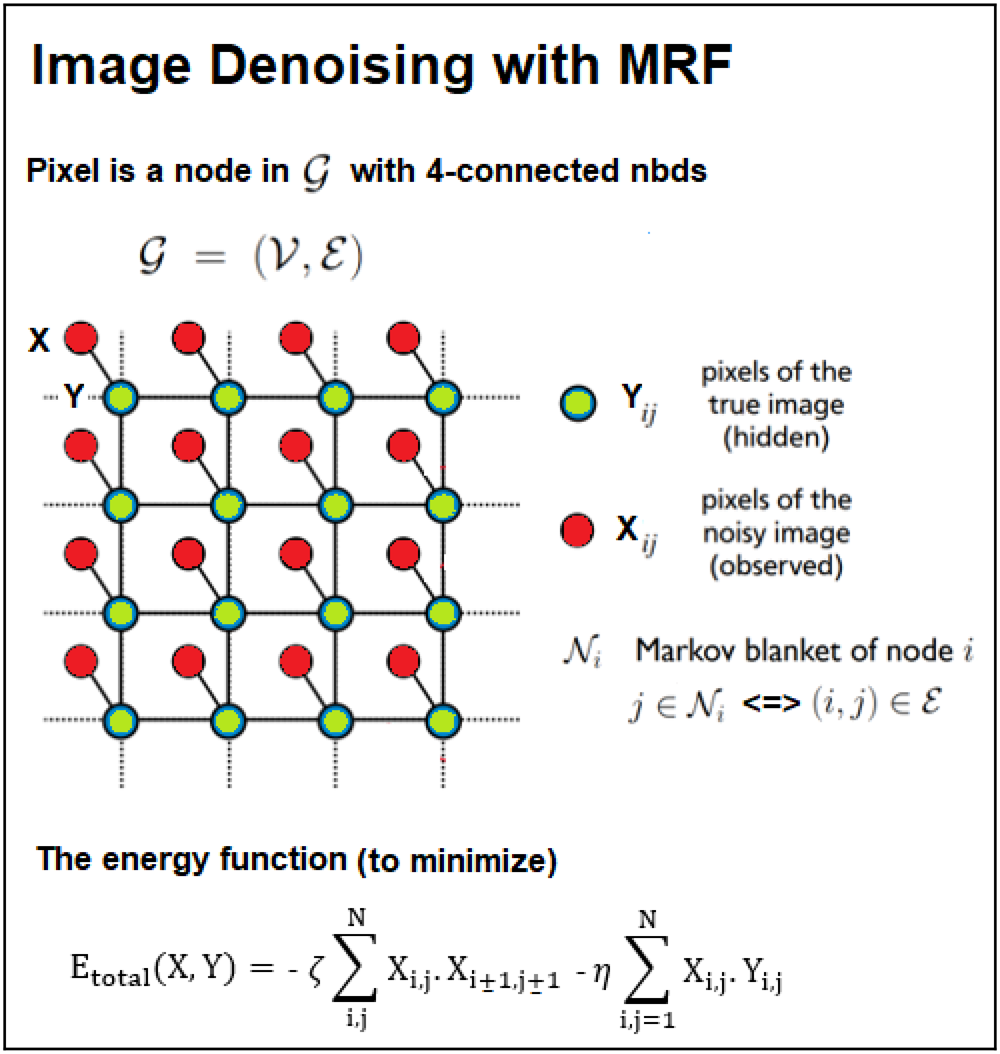
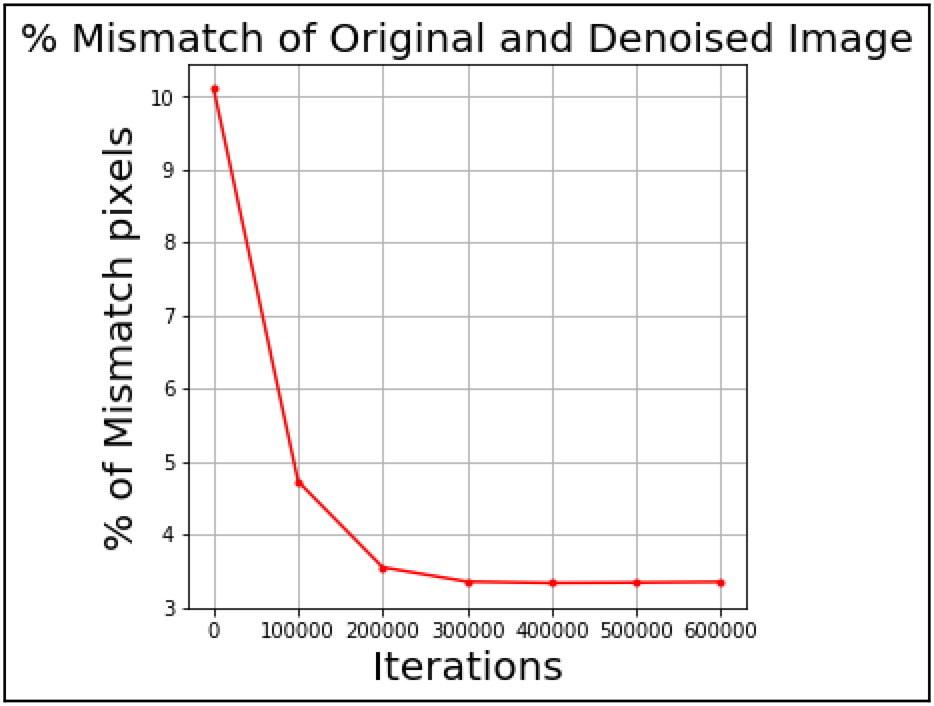
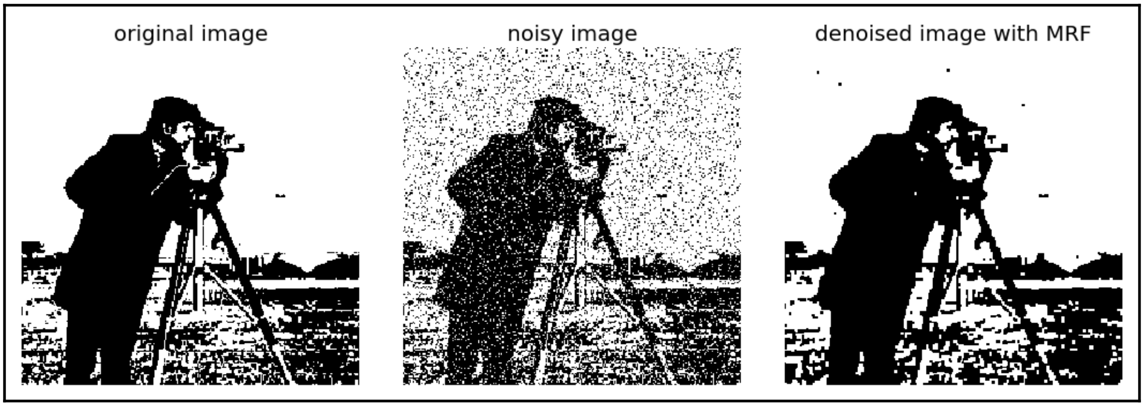
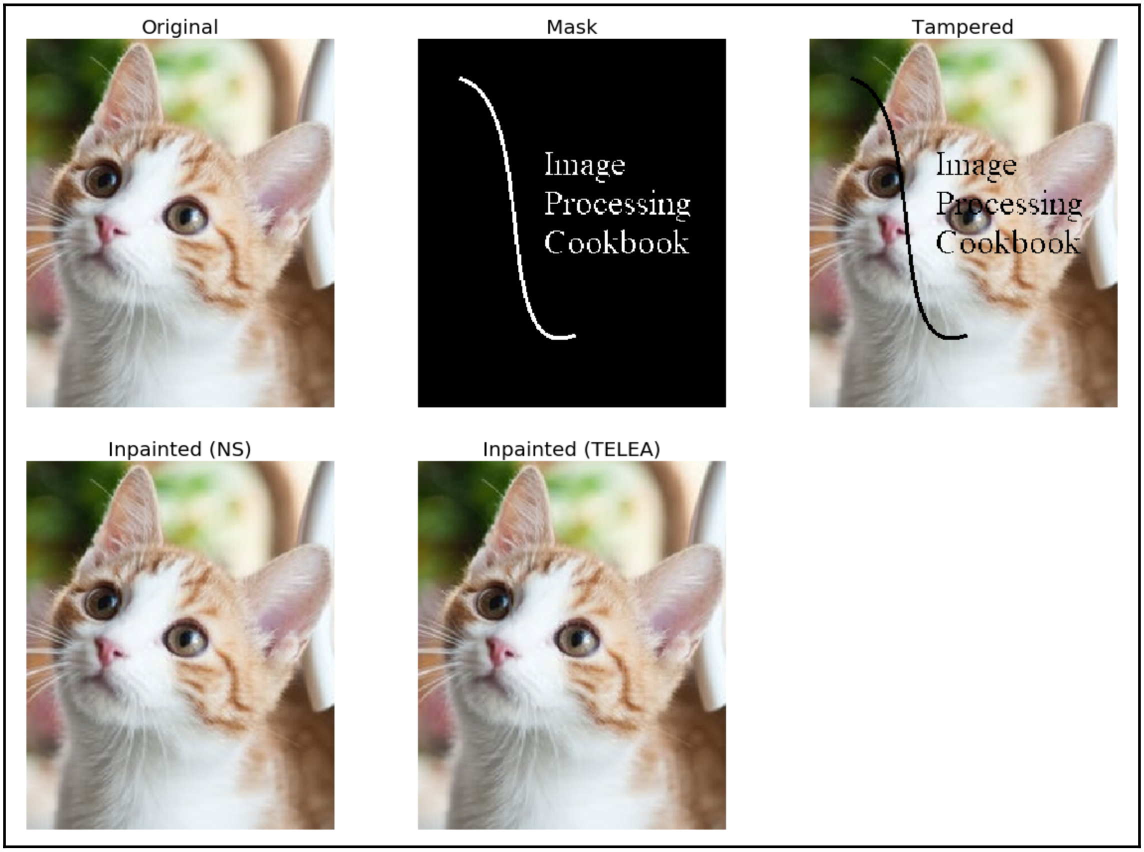
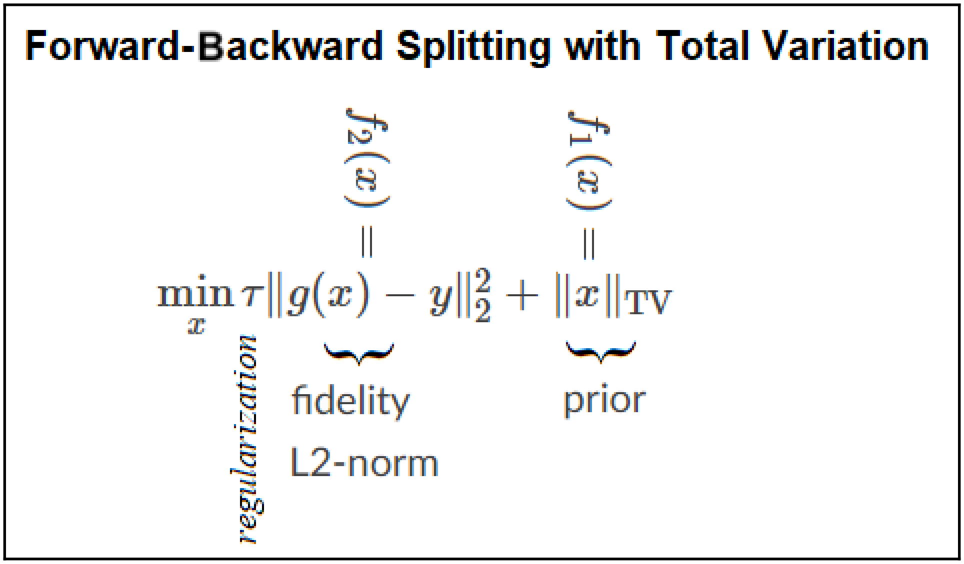


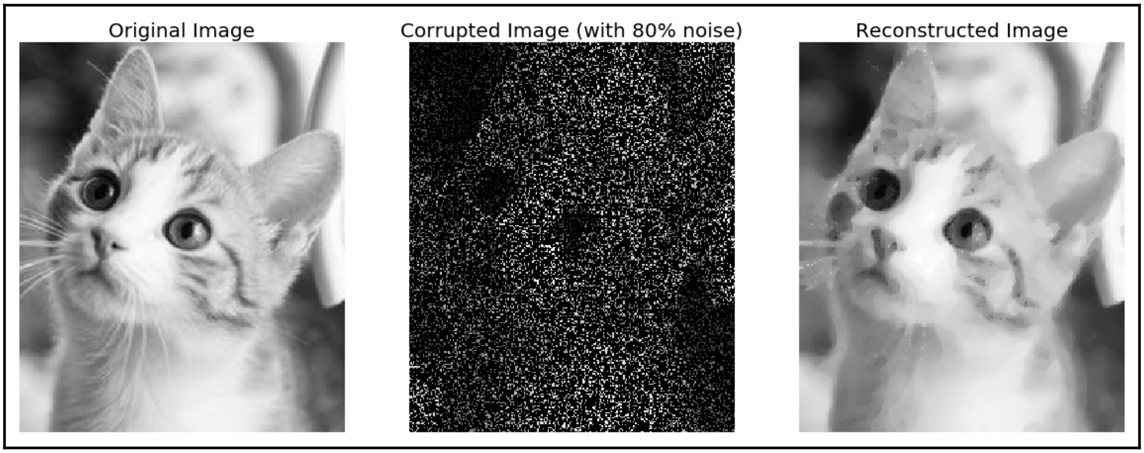
Image Denoising with MRF
Pixel is a node in $\mathcal{G}$ with 4-connected nbds
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$
X
Y
Y_{ij}
pixels of the true image (hidden)
X_{ij}
pixels of the noisy image (observed)
$\mathcal{N}_i$ Markov blanket of node i
$j \in \mathcal{N}_i$ <=> $(i, j) \in \mathcal{E}$
The energy function (to minimize)
$E_{total}(X, Y) = -\zeta \sum_{i,j}^{N} X_{i,j} . X_{i\pm1,j\pm1} - \eta \sum_{i,j=1}^{N} X_{i,j} . Y_{i,j}$


% Mismatch of Original and Denoised Image
% of Mismatch pixels
Iterations
10
9
8
7
6
5
4
3
0
100000
200000
300000
400000
500000
600000

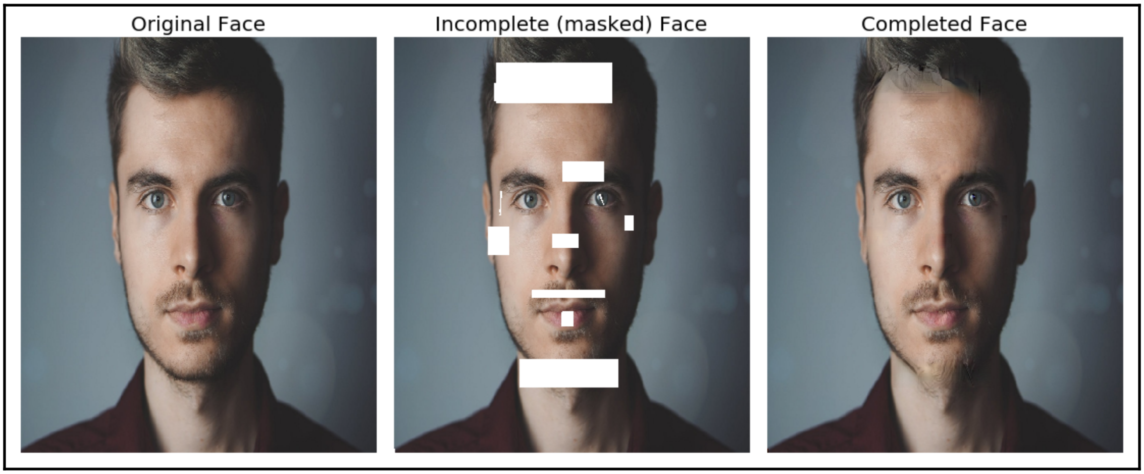

original image
noisy image
denoised image with MRF

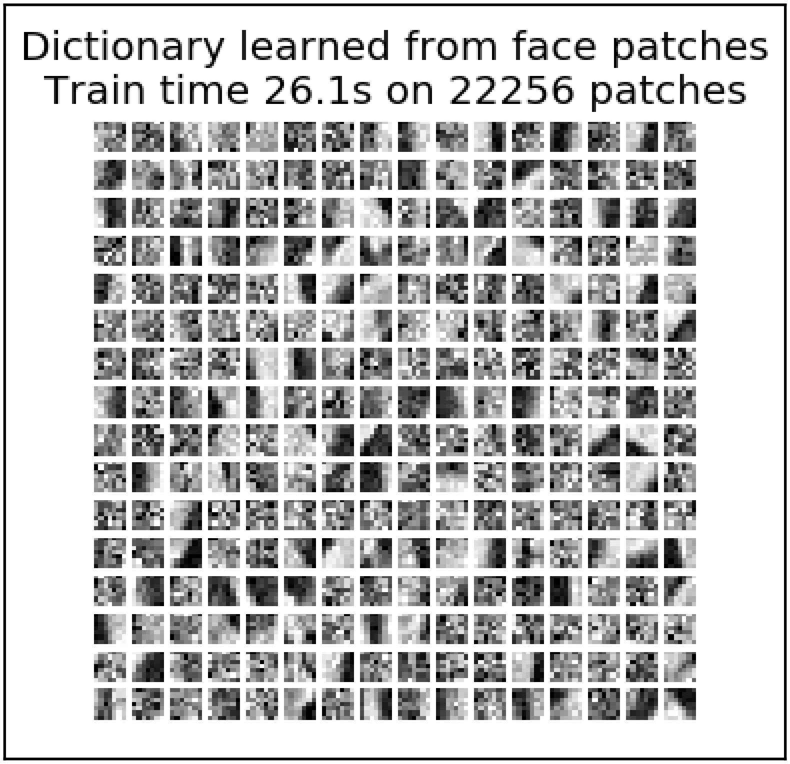

Original
Mask
Image
Processing
Cookbook
Tampered
Image
Processing
Cookbook
Inpainted (NS)
Inpainted (TELEA)


Forward-Backward Splitting with Total Variation
$f_2(x)$
$f_1(x)$
$\min_x \tau \|g(x) - y\|_2^2 + \|x\|_{\mathrm{TV}}$
regularization
fidelity
L2-norm
prior

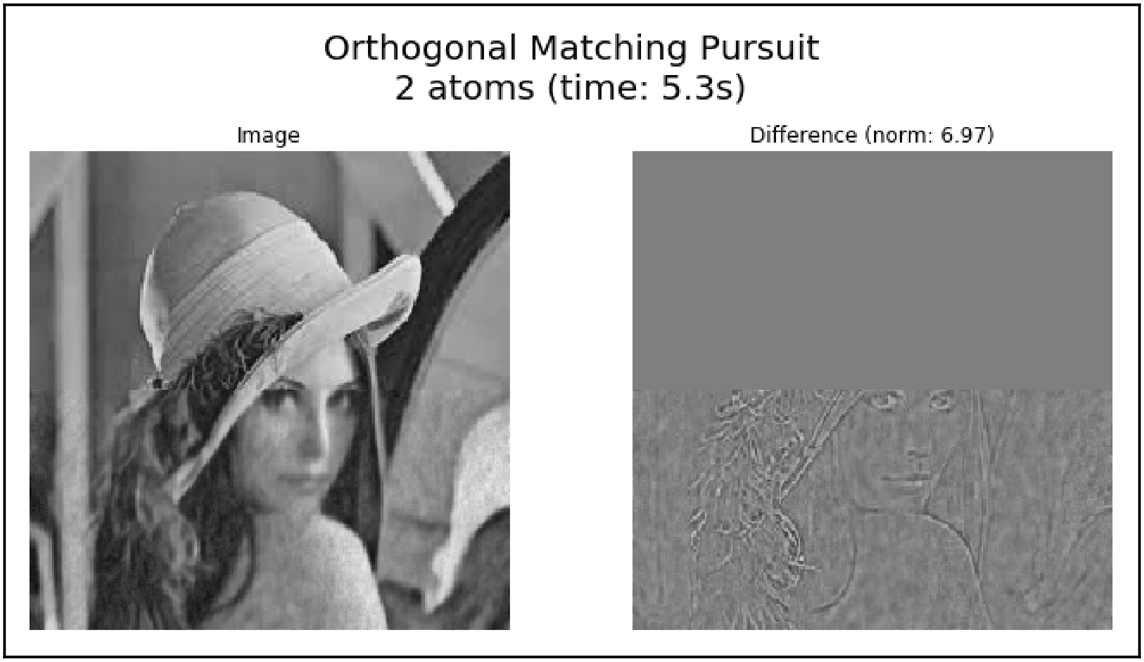

Original Image
Corrupted Image (with 80% noise)
Reconstructed Image

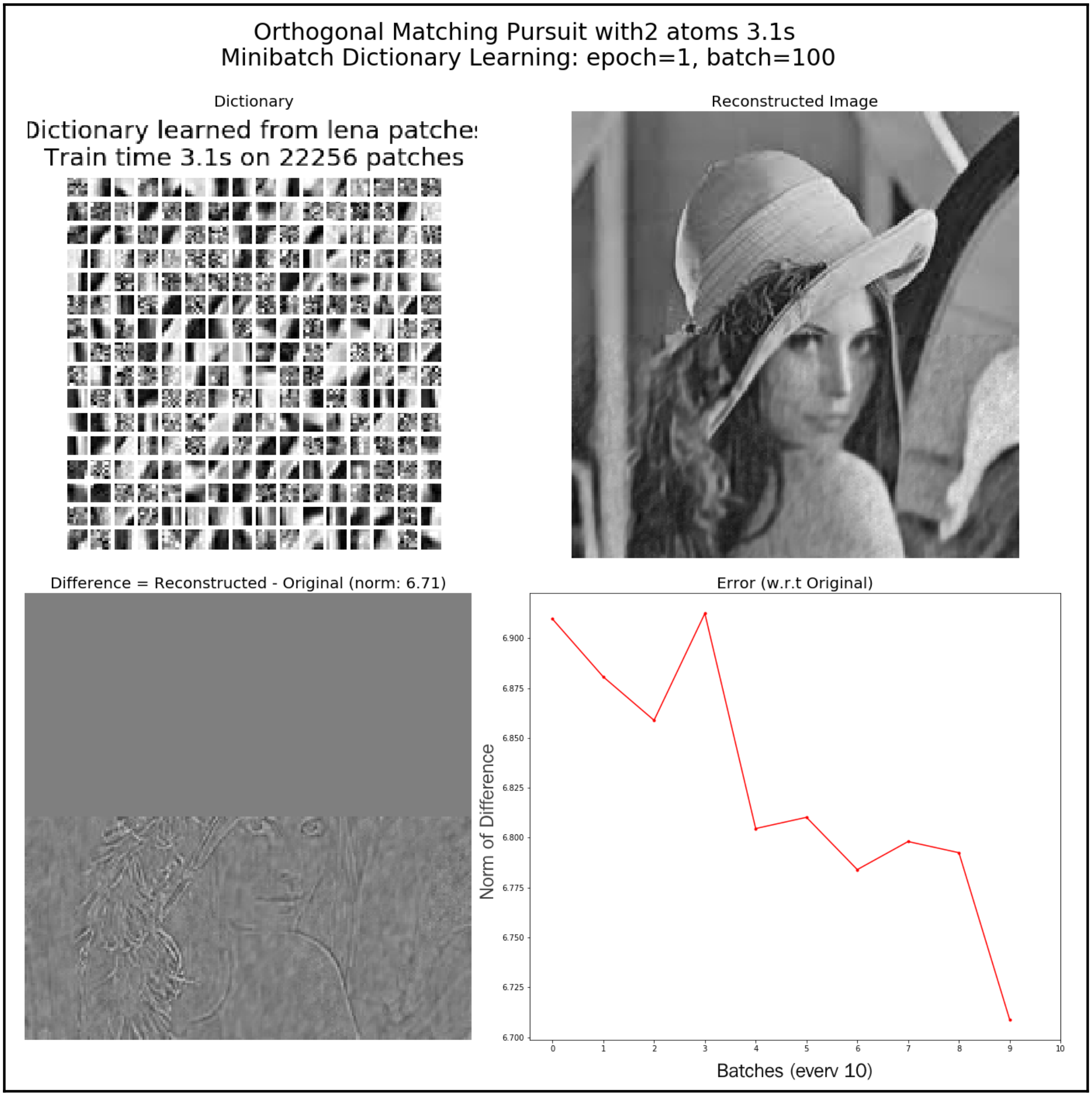

Incomplete Image
Completed Image (with Completion Net)


Original Face
Incomplete (masked) Face
Completed Face


Dictionary learned from face patches
Train time 26.1s on 22256 patches

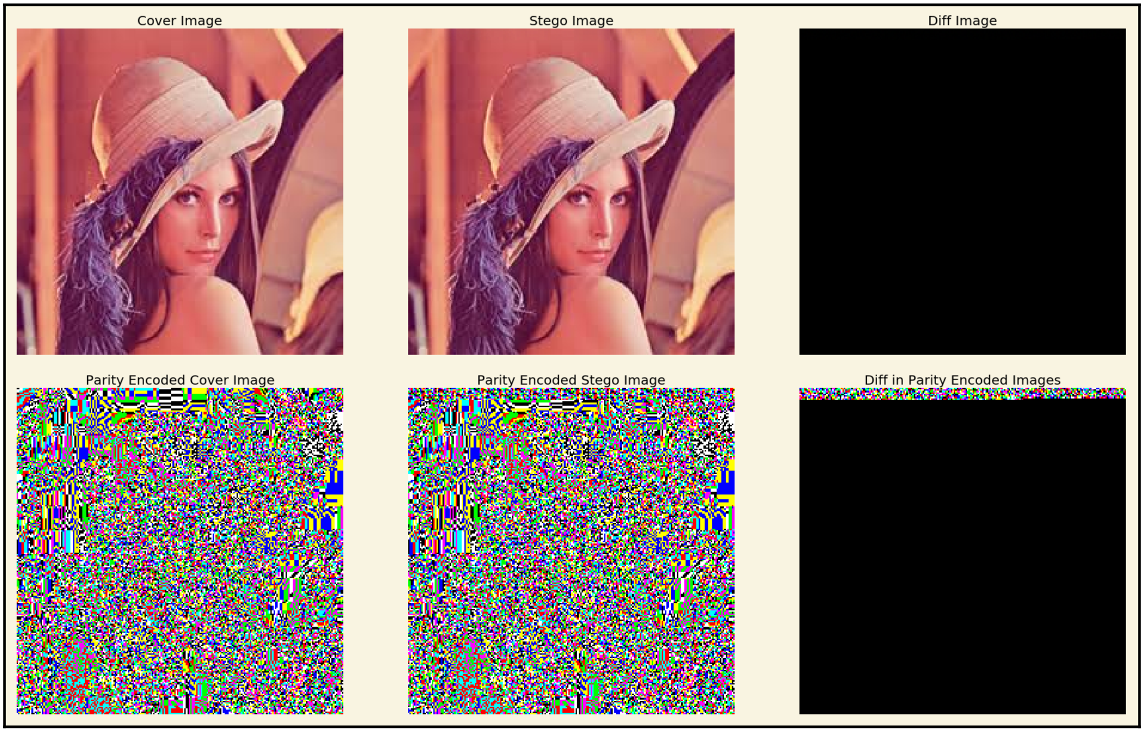

Distorted image
Image
Difference (norm: 13.24)

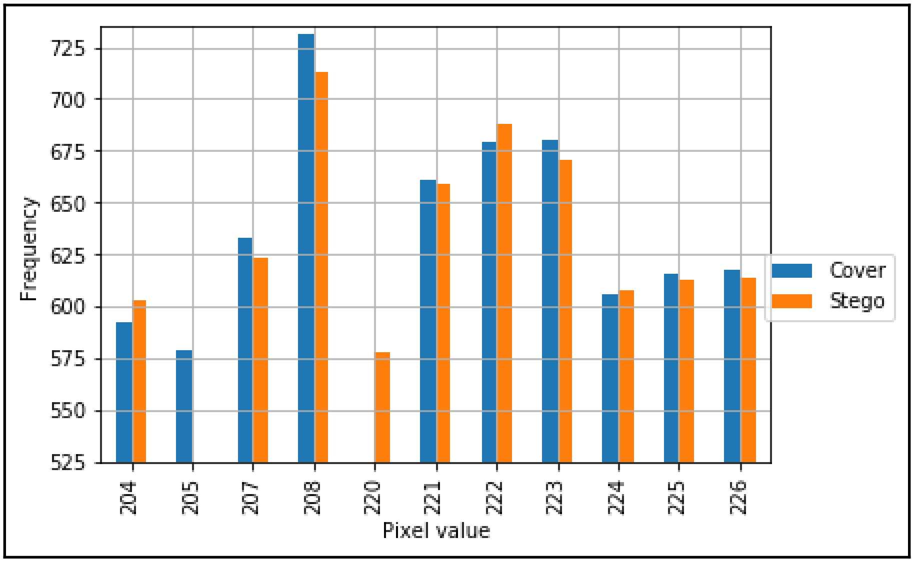

Orthogonal Matching Pursuit
2 atoms (time: 5.3s)
Image
Difference (norm: 6.97)

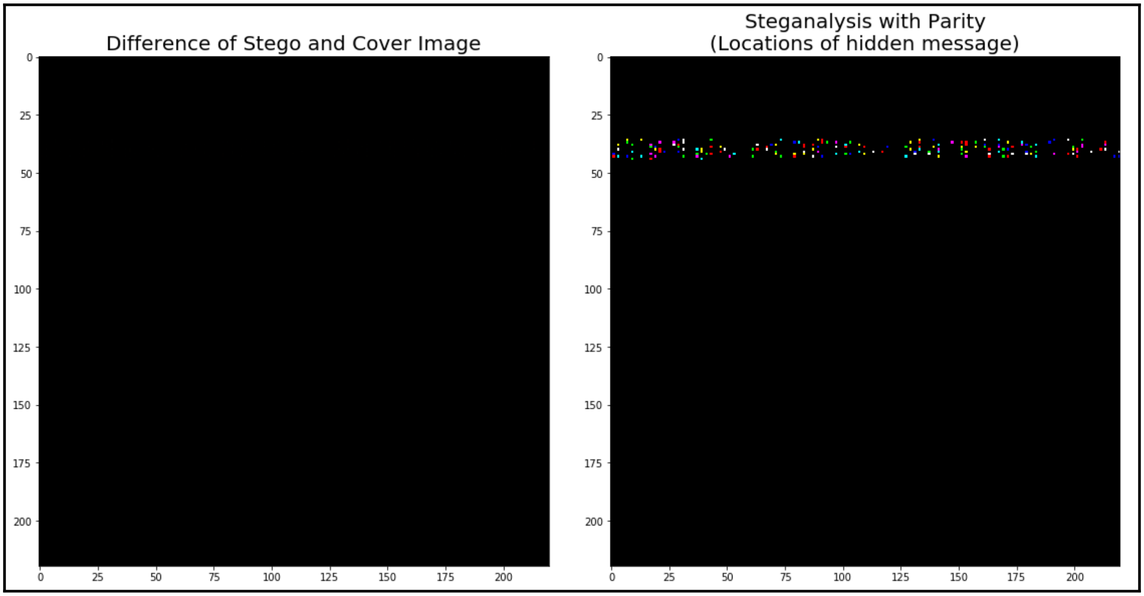


Orthogonal Matching Pursuit with2 atoms 3.1s
Minibatch Dictionary Learning: epoch=1, batch=100
Dictionary
Dictionary learned from lena patche:
Train time 3.1s on 22256 patches
Reconstructed Image
Difference = Reconstructed - Original (norm: 6.71)
Error (w.r.t Original)
Norm of Difference
Batches (everv 10)


original image, zeros=0.215%
image (div 8), zeros=1.788%
image with DWT (D8)
zeros in transform=88.924%
image with DWT (D8) + soft thresholding
zeros in transform=94.547%


Cover Image
Stego Image
Diff Image
Parity Encoded Cover Image
Parity Encoded Stego Image
Diff in Parity Encoded Images


725
700
675
650
625
600
575
550
525
Frequency
Cover
Stego
204
205
207
208
220
221
222
223
224
225
226
Pixel value


Difference of Stego and Cover Image
Steganalysis with Parity
(Locations of hidden message)
0
25
50
75
100
125
150
175
200

Chapter 4: Binary Image Processing

Erosion $A \ominus B = \{z \mid (B)_z \subseteq A\}$

shrink the (foreground) object

Dilation $A \oplus B = \{z \mid (\hat{B})_z \cap A \neq \Phi\}$

grow the (foreground) object

Fundamental Morphological Operations

Opening $A \circ B = (A \ominus B) \oplus B$

Closing $A \bullet B = (A \oplus B) \ominus B$

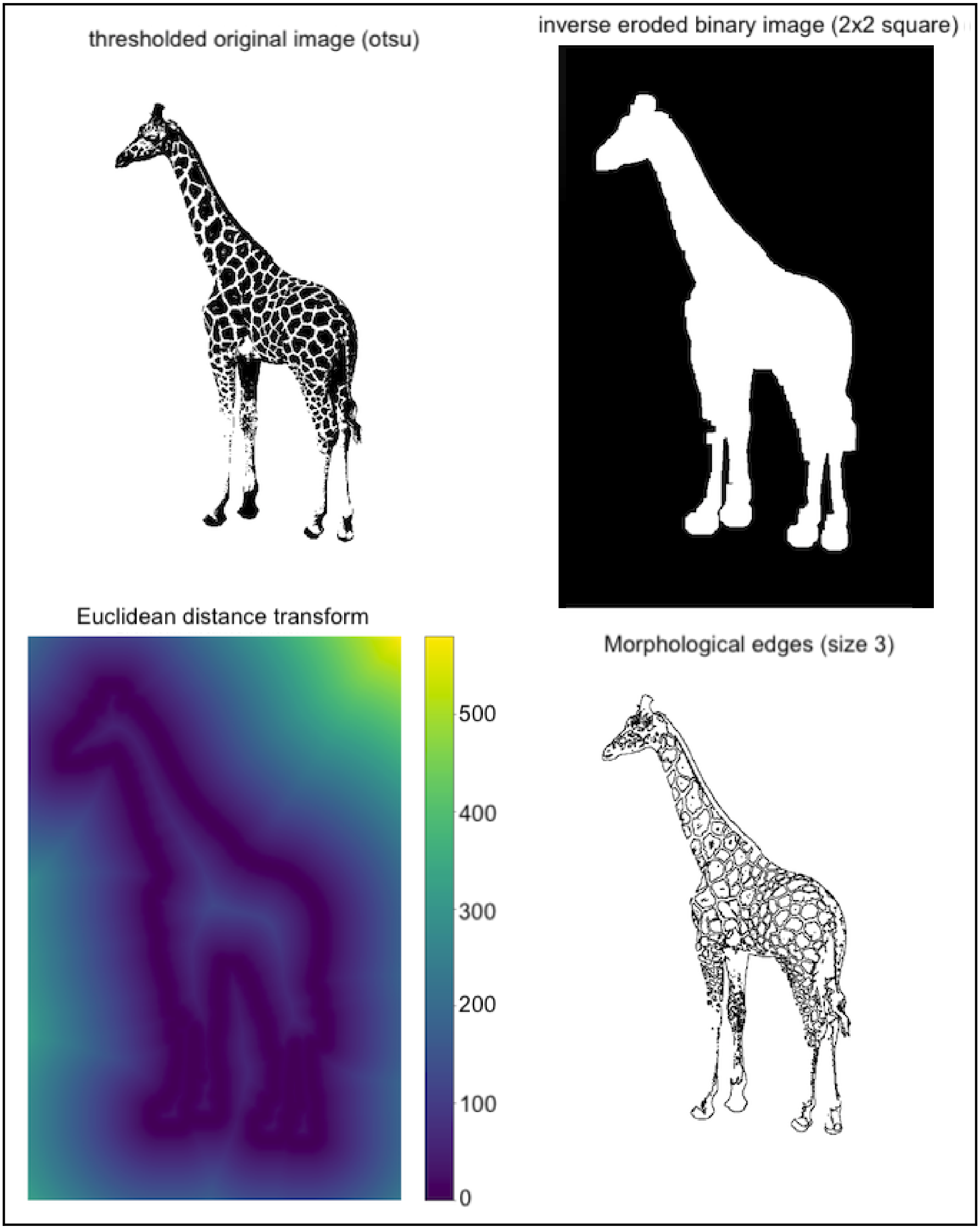
Hit or Miss Transform $A \circledast \{B_1, B_2\} = (A \ominus B_1) \cap (A^c \ominus B_2)$

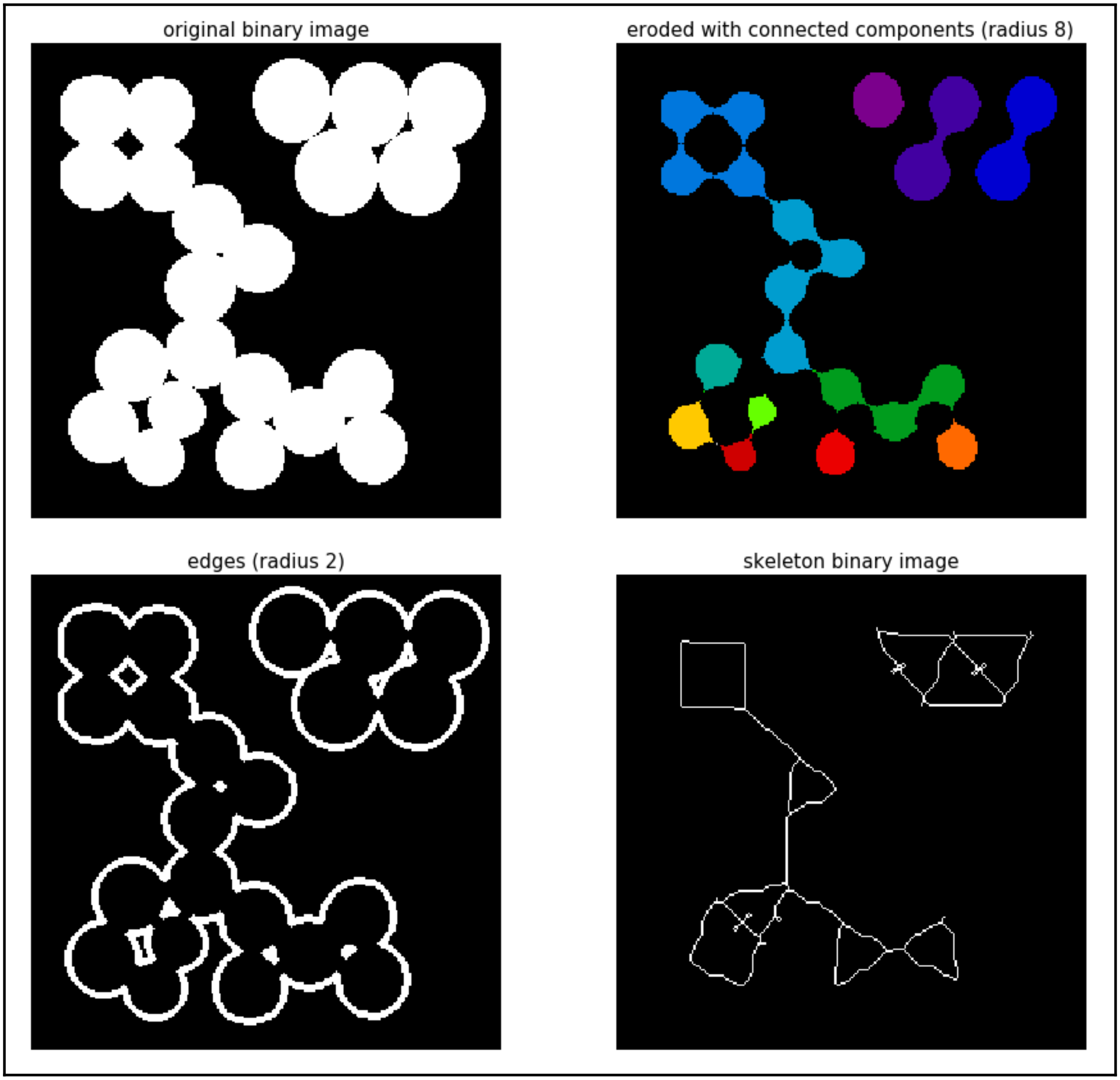
Compound Morphological Operations

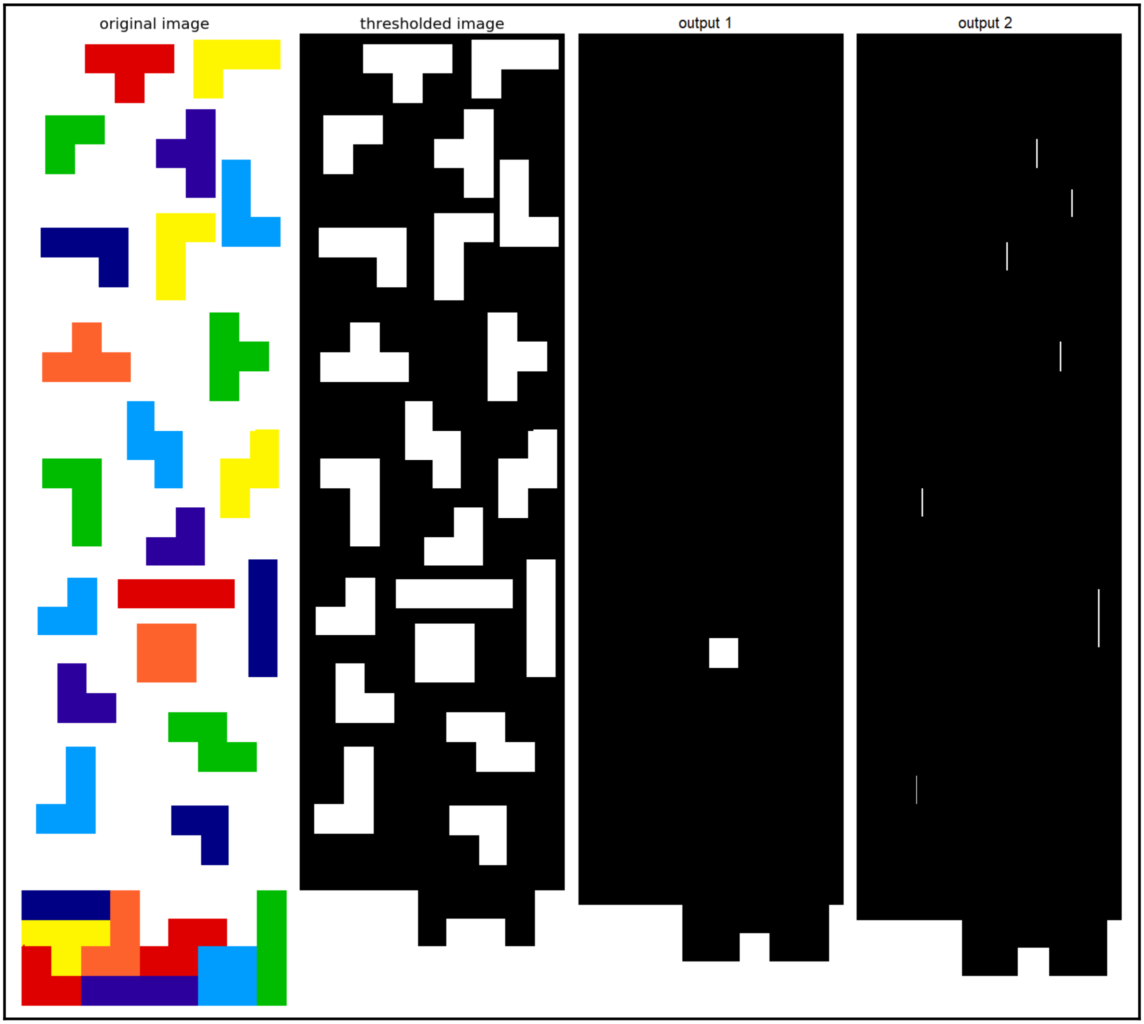
Duality $(A \bullet B)^c = (A^c \circ \hat{B})$

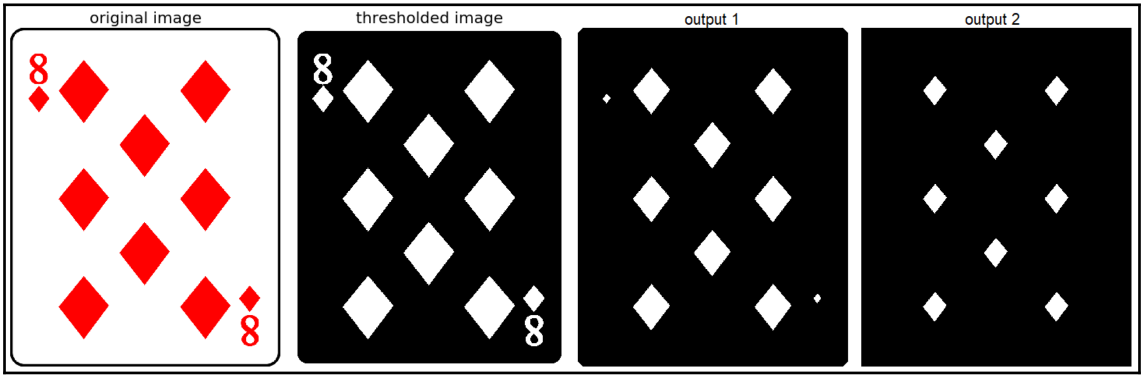
Binary Image (Set)

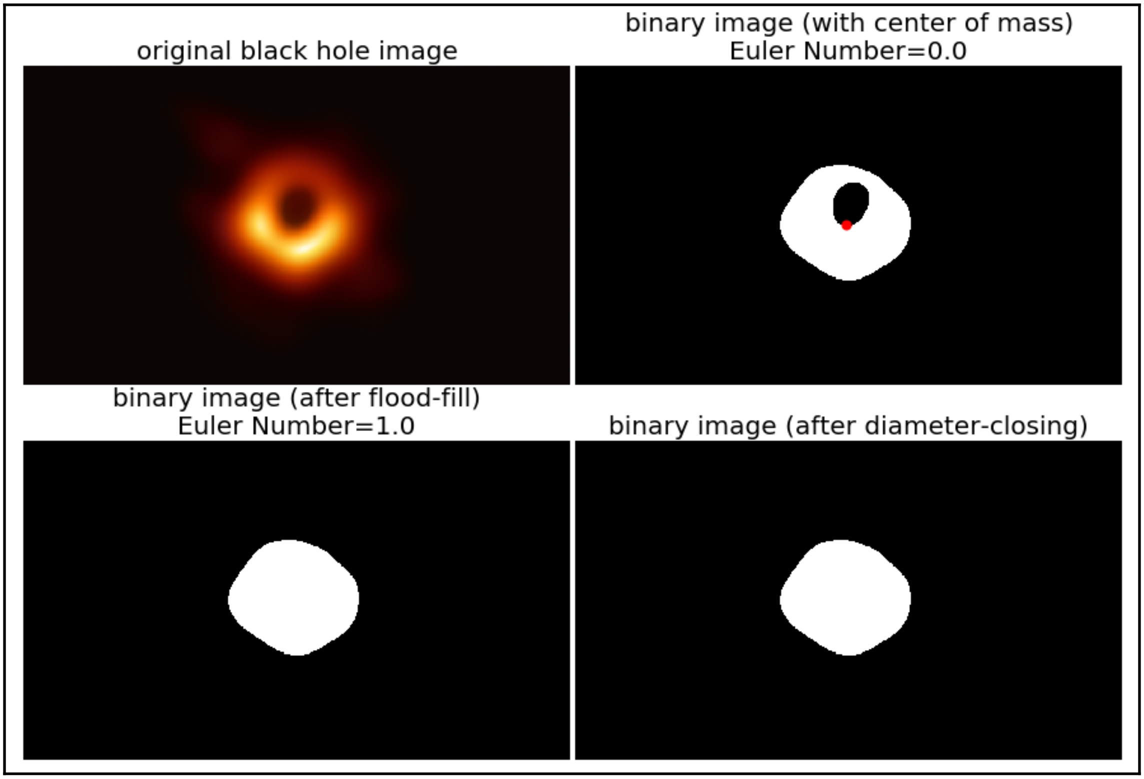
Structuring Element (SE)

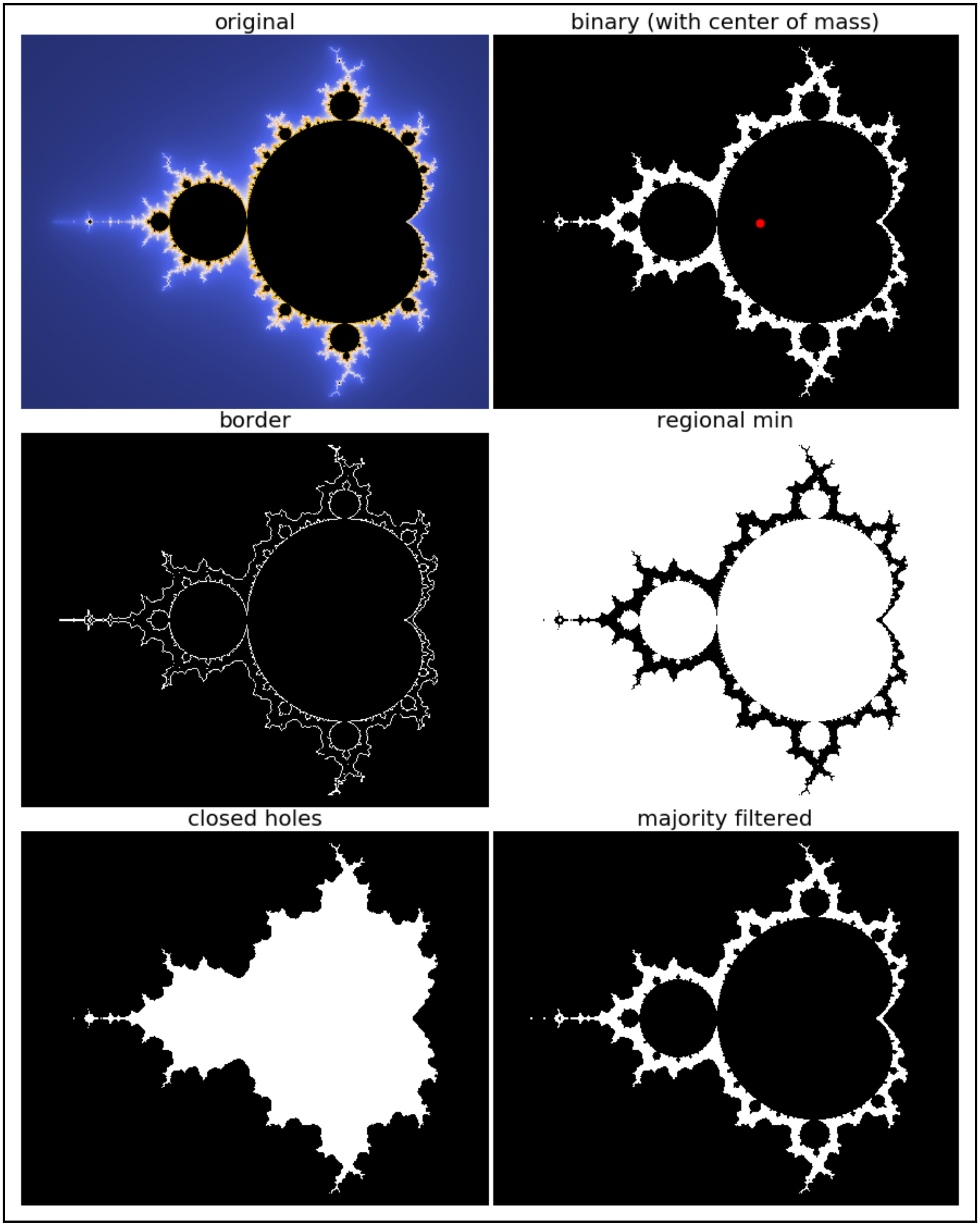

thresholded original image (otsu)
inverse eroded binary image (2x2 square)
Euclidean distance transform
500
400
300
200
100
0
Morphological edges (size 3)

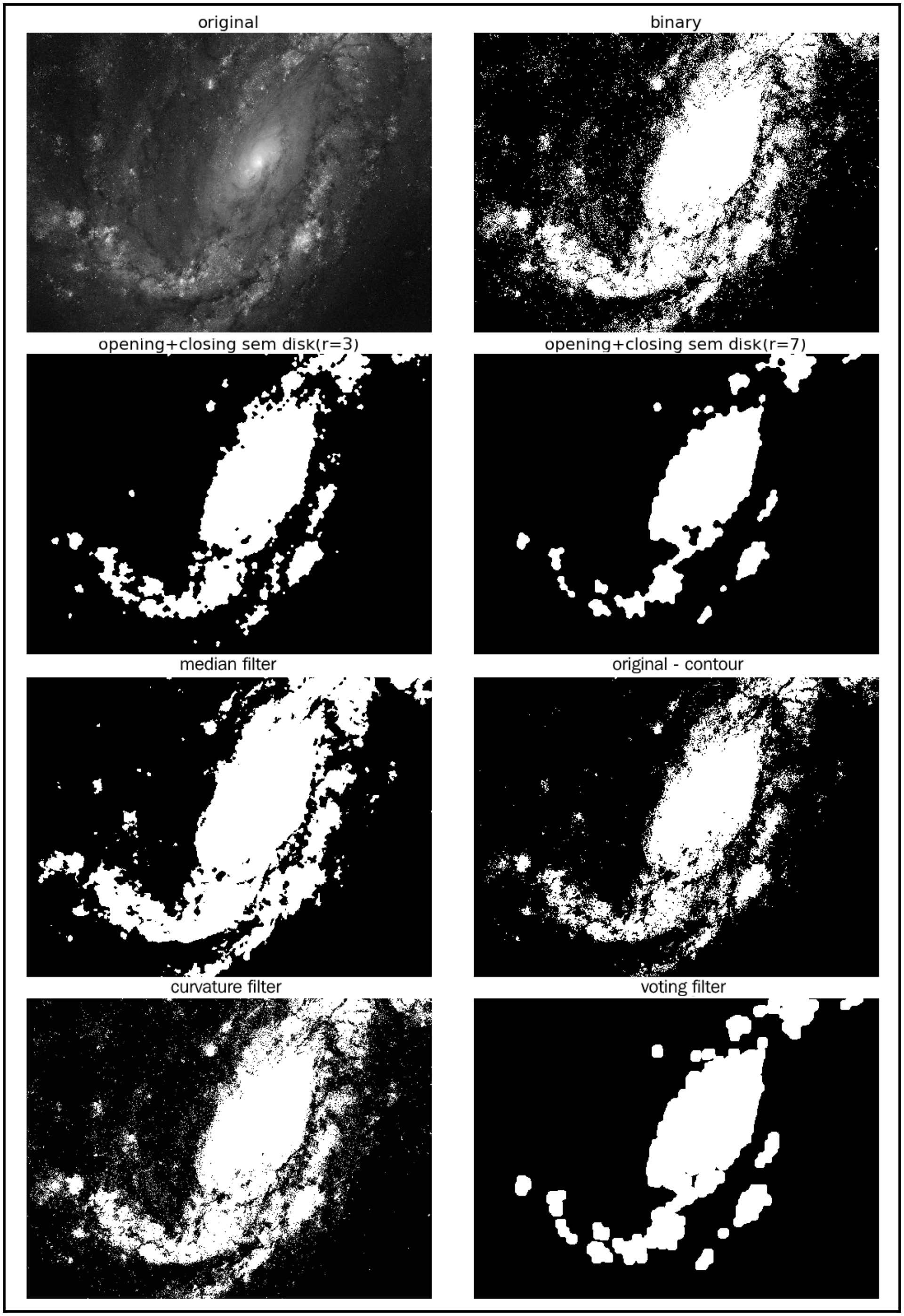

original binary image
eroded with connected components (radius 8)
edges (radius 2)
skeleton binary image


original image
thresholded image
output 1
output 2

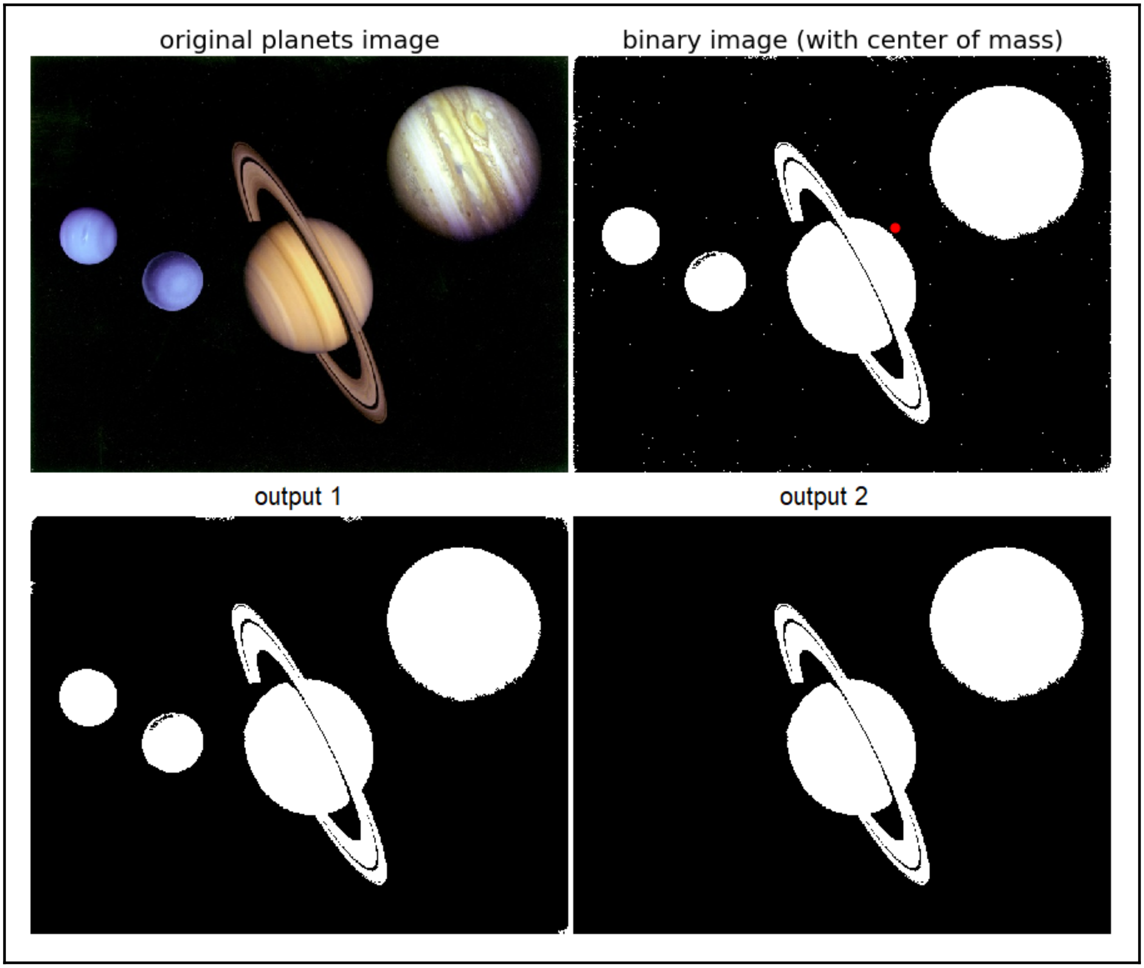

original image
thresholded image
output 1
output 2

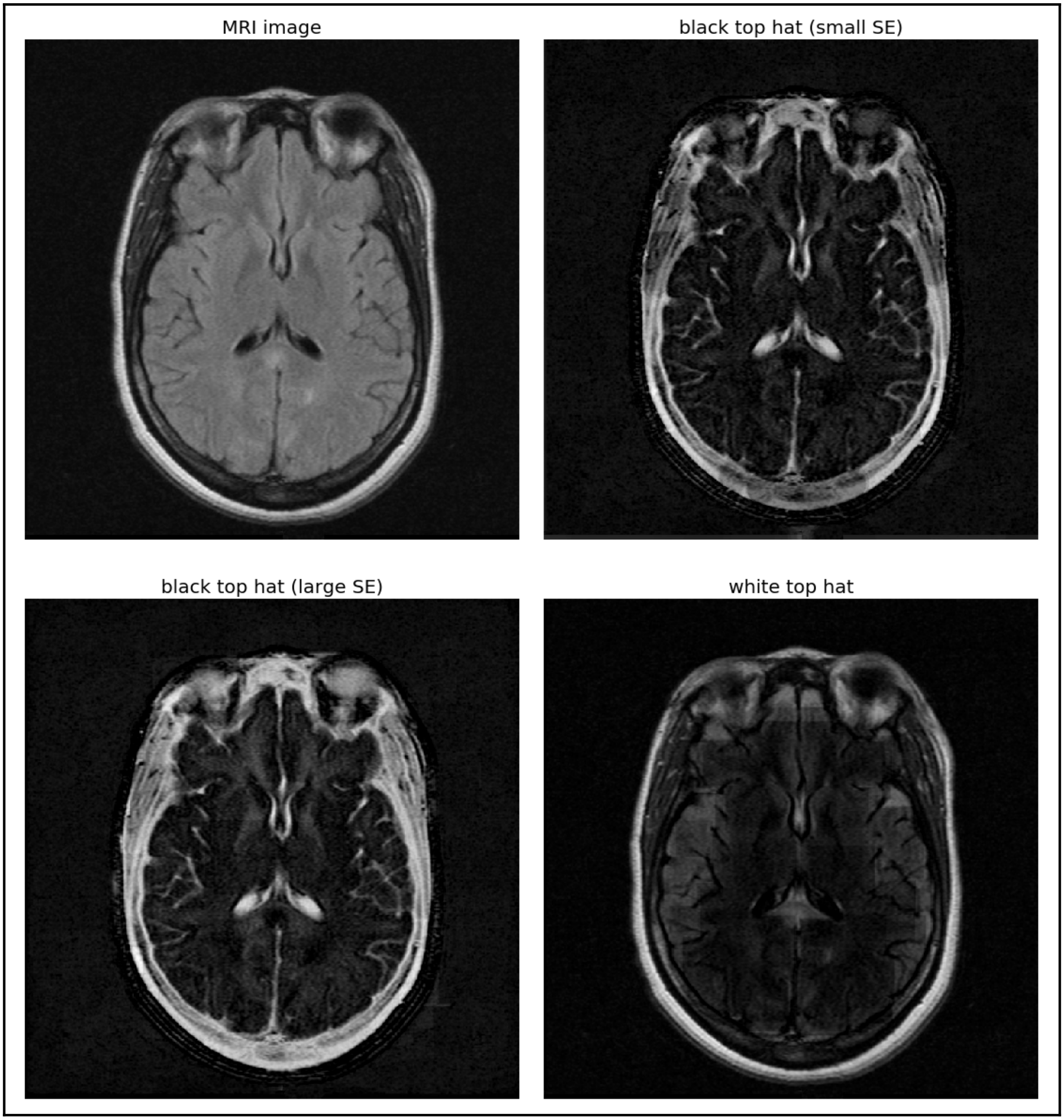

original black hole image
binary image (with center of mass)
Euler Number=0.0
binary image (after flood-fill)
Euler Number=1.0
binary image (after diameter-closing)

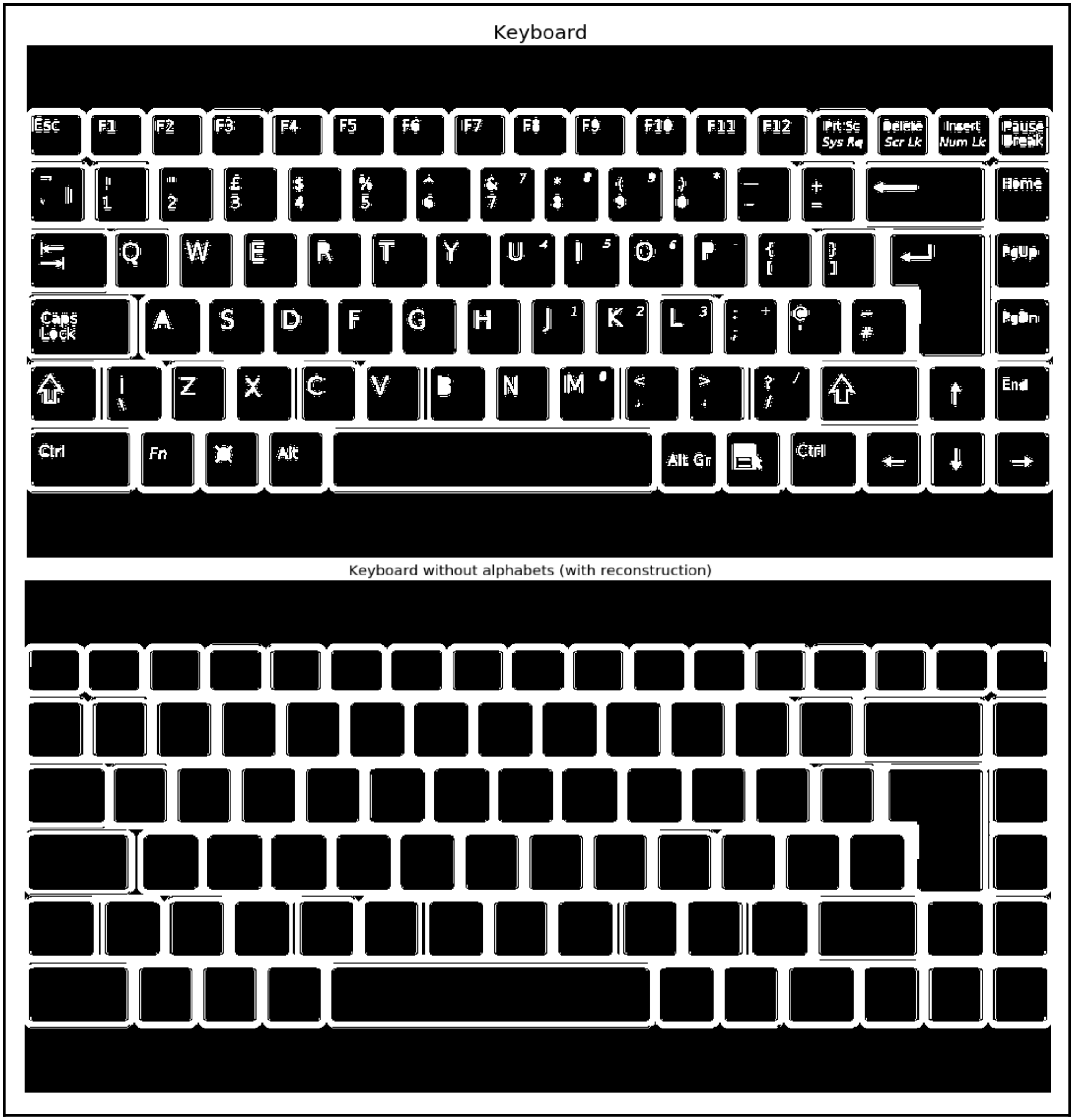

original
binary (with center of mass)
border
regional min
closed holes
majority filtered

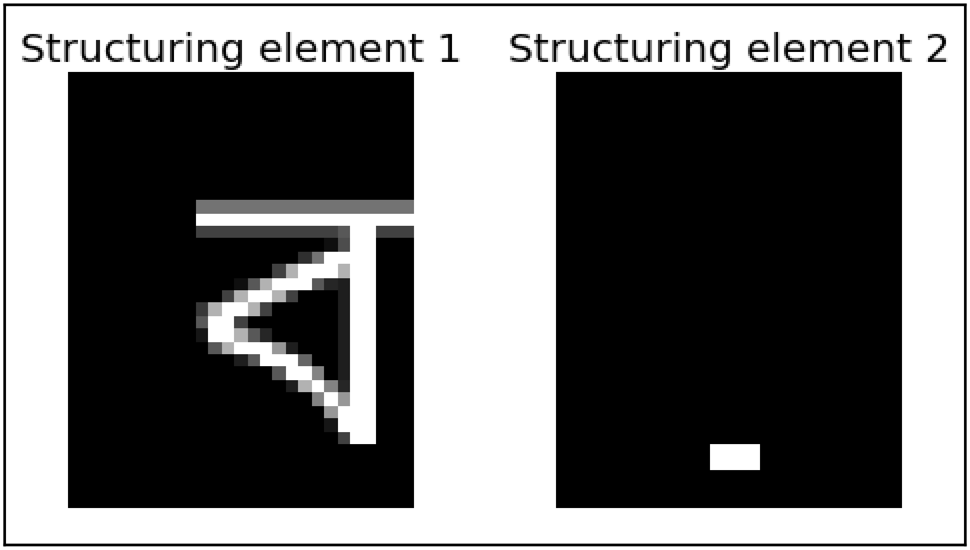

original
binary
opening+closing sem disk(r=3)
opening+closing sem disk(r=7)
median filter
original - contour
curvature filter
voting filter

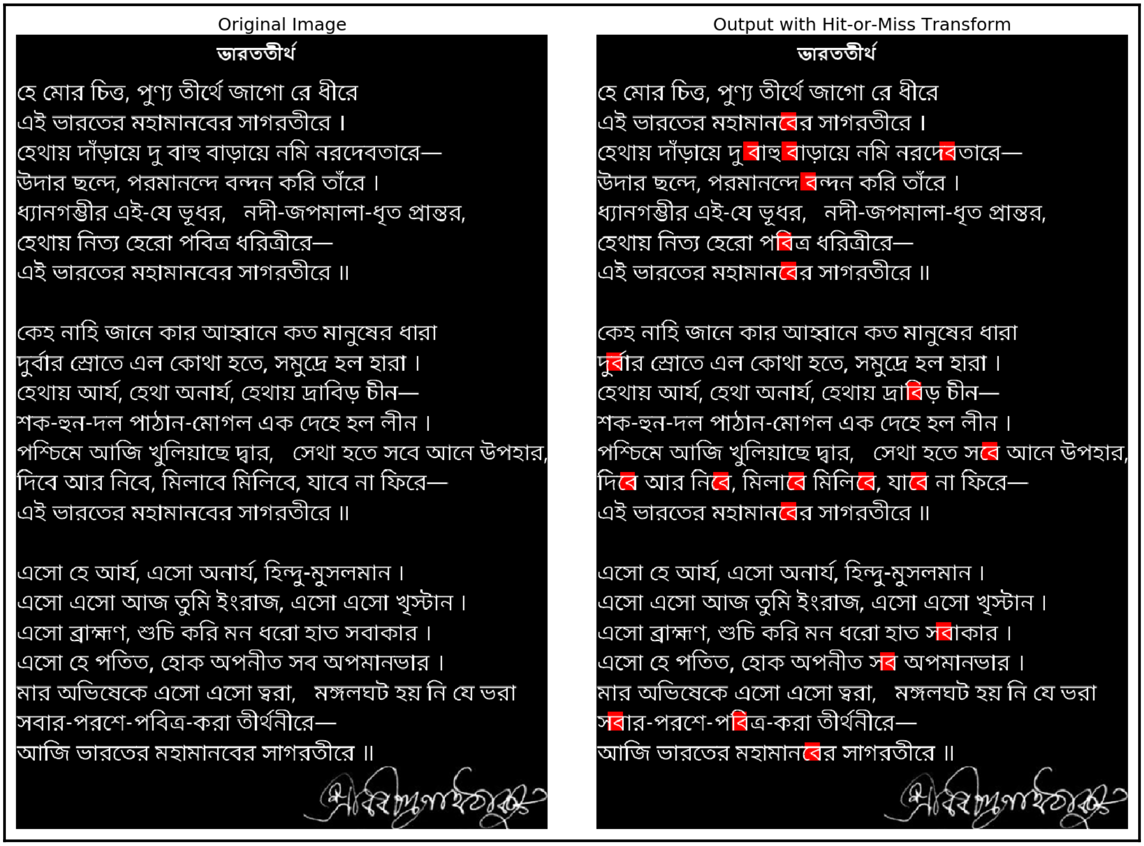

Tagore's drawing-ridden manuscript
Seed for the reconstruction
Tagore's drawing
Tagore's manuscript without drawing

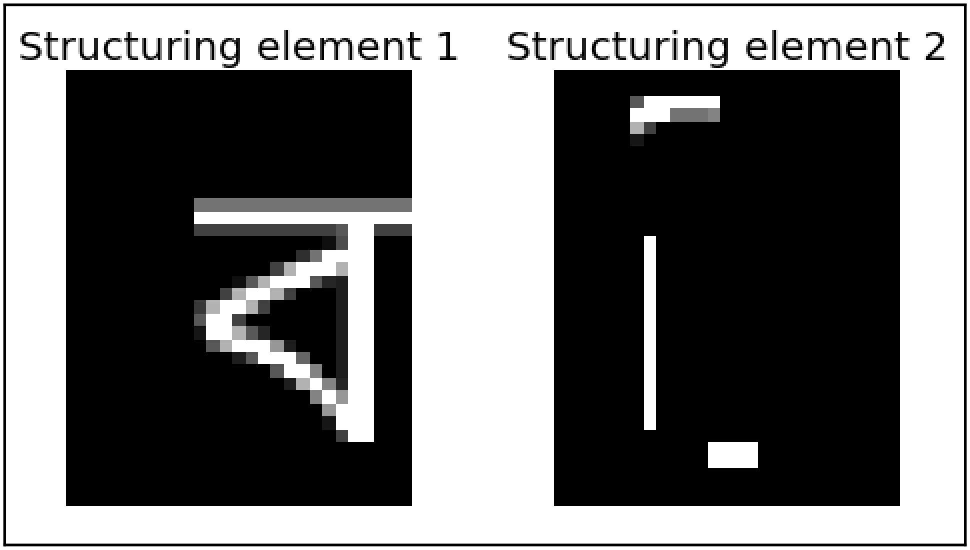

original planets image
binary image (with center of mass)
output 1
output 2

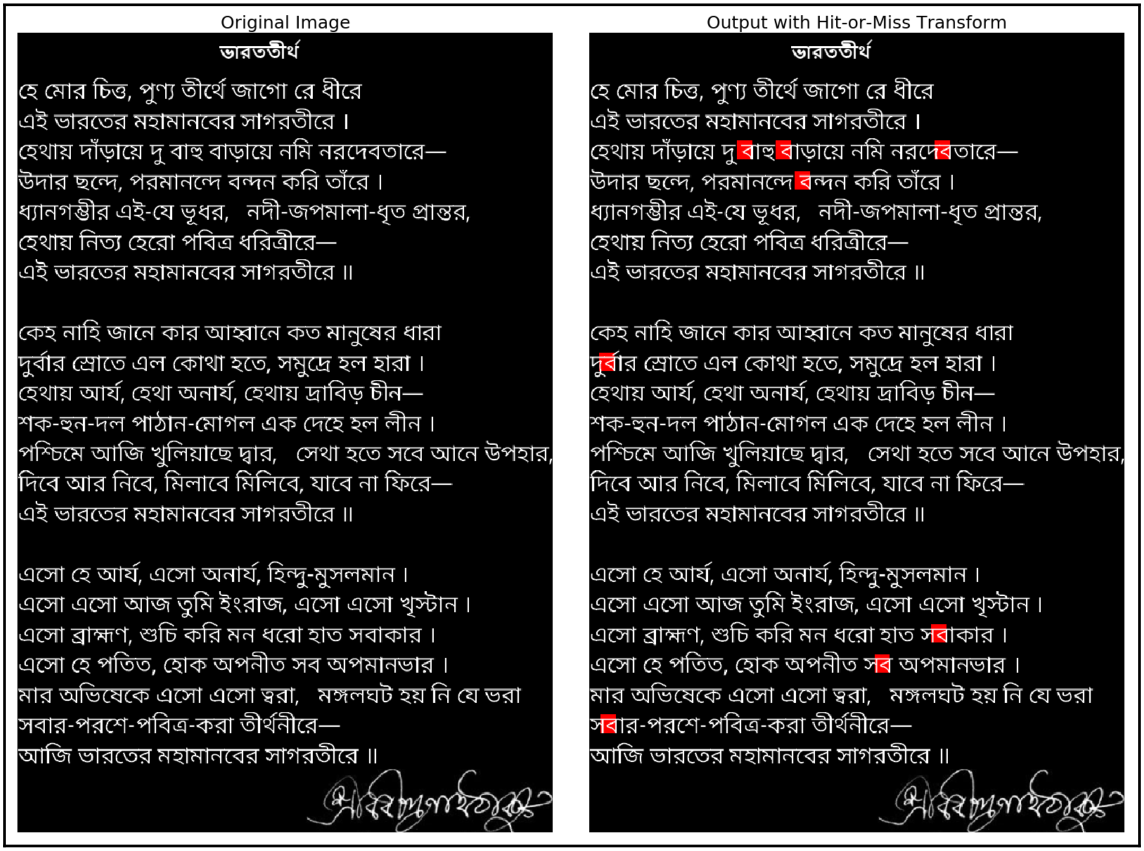

MRI image
black top hat (small SE)
black top hat (large SE)
white top hat


Keyboard
Keyboard without alphabets (with reconstruction)

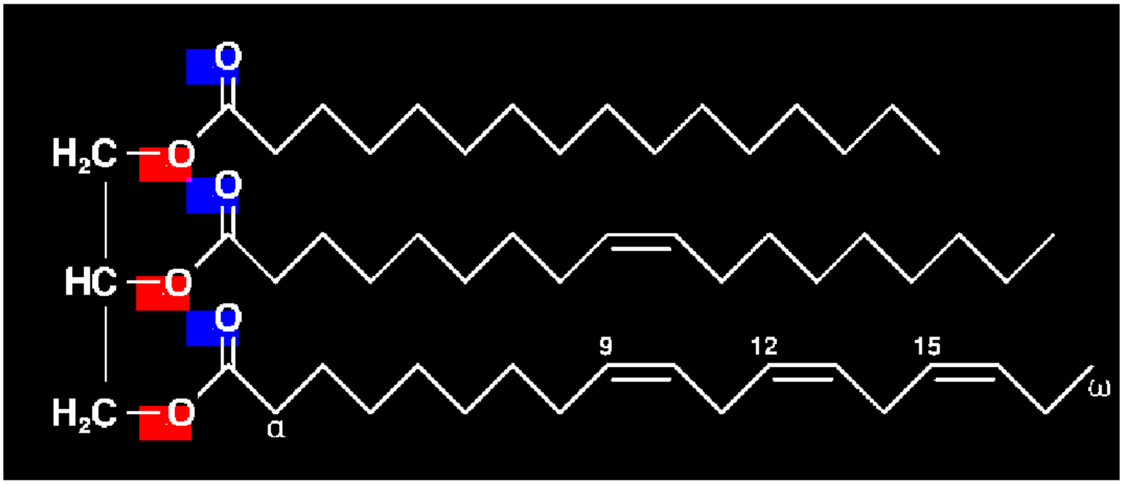

Structuring element 1
Structuring element 2

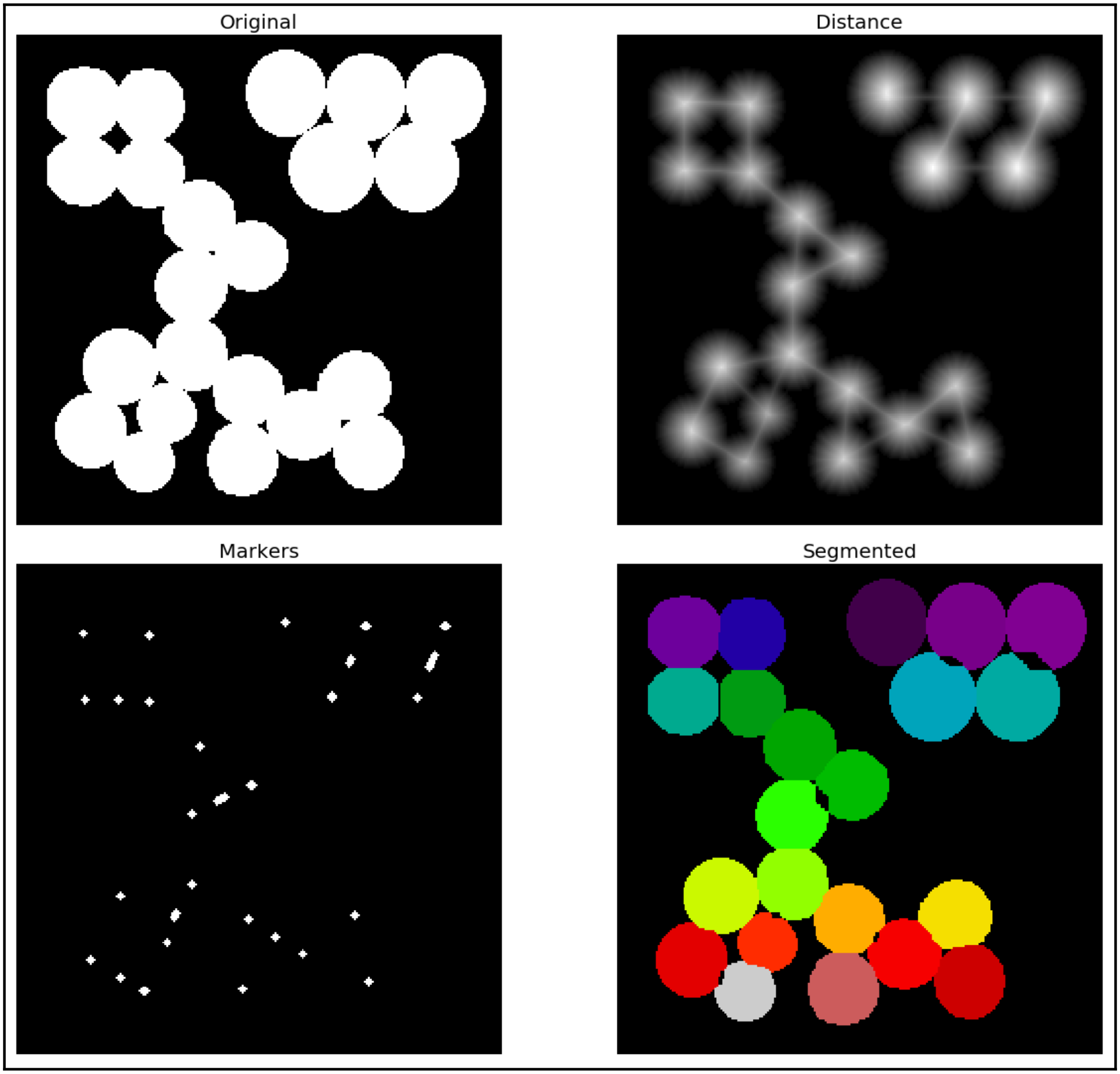

Original Image
ভারততীর্থ
হে মোর চিত্ত, পুণ্য তীর্থে জাগো রে ধীরে
এই ভারতের মহামানবের সাগরতীরে ।
হেথায় দাঁড়ায়ে দু বাহু বাড়ায়ে নমি নরদেবতারে—
উদার ছন্দে, পরমানন্দে বন্দন করি তাঁরে ।
ধ্যানগম্ভীর এই-যে ভূধর, নদী-জপমালা-ধৃত প্রান্তর,
হেথায় নিত্য হেরো পবিত্র ধরিত্রীরে—
এই ভারতের মহামানবের সাগরতীরে ॥
কেহ নাহি জানে কার আহ্বানে কত মানুষের ধারা
দুর্বার স্রোতে এল কোথা হতে, সমুদ্রে হল হারা ।
হেথায় আর্য, হেথা অনার্য, হেথায় দ্রাবিড় চীন—
শক-হূন-দল পাঠান-মোগল এক দেহে হল লীন ।
পশ্চিমে আজি খুলিয়াছে দ্বার, সেথা হতে সবে আনে উপহার,
দিবে আর নিবে, মিলাবে মিলিবে, যাবে না ফিরে—
এই ভারতের মহামানবের সাগরতীরে ॥
এসো হে আর্য, এসো অনার্য, হিন্দু-মুসলমান ।
এসো এসো আজ তুমি ইংরাজ, এসো এসো খৃস্টান ।
এসো ব্রাহ্মণ, শুচি করি মন ধরো হাত সবাকার ।
এসো হে পতিত, হোক অপনীত সব অপমানভার ।
মার অভিষেকে এসো এসো ত্বরা, মঙ্গলঘট হয় নি যে ভরা
সবার-পরশে-পবিত্র-করা তীর্থনীরে—
আজি ভারতের মহামানবের সাগরতীরে ॥
Output with Hit-or-Miss Transform
ভারততীর্থ
হে মোর চিত্ত, পুণ্য তীর্থে জাগো রে ধীরে
এই ভারতের মহামানবের সাগরতীরে ।
হেথায় দাঁড়ায়ে দু বাহু বাড়ায়ে নমি নরদেবতারে—
উদার ছন্দে, পরমানন্দে বন্দন করি তাঁরে ।
ধ্যানগম্ভীর এই-যে ভূধর, নদী-জপমালা-ধৃত প্রান্তর,
হেথায় নিত্য হেরো পবিত্র ধরিত্রীরে—
এই ভারতের মহামানবের সাগরতীরে ॥
কেহ নাহি জানে কার আহ্বানে কত মানুষের ধারা
দুর্বার স্রোতে এল কোথা হতে, সমুদ্রে হল হারা ।
হেথায় আর্য, হেথা অনার্য, হেথায় দ্রাবিড় চীন—
শক-হূন-দল পাঠান-মোগল এক দেহে হল লীন ।
পশ্চিমে আজি খুলিয়াছে দ্বার, সেথা হতে সবে আনে উপহার,
দিবে আর নিবে, মিলাবে মিলিবে, যাবে না ফিরে—
এই ভারতের মহামানবের সাগরতীরে ॥
এসো হে আর্য, এসো অনার্য, হিন্দু-মুসলমান ।
এসো এসো আজ তুমি ইংরাজ, এসো এসো খৃস্টান ।
এসো ব্রাহ্মণ, শুচি করি মন ধরো হাত সবাকার ।
এসো হে পতিত, হোক অপনীত সব অপমানভার ।
মার অভিষেকে এসো এসো ত্বরা, মঙ্গলঘট হয় নি যে ভরা
সবার-পরশে-পবিত্র-করা তীর্থনীরে—
আজি ভারতের মহামানবের সাগরতীরে ॥

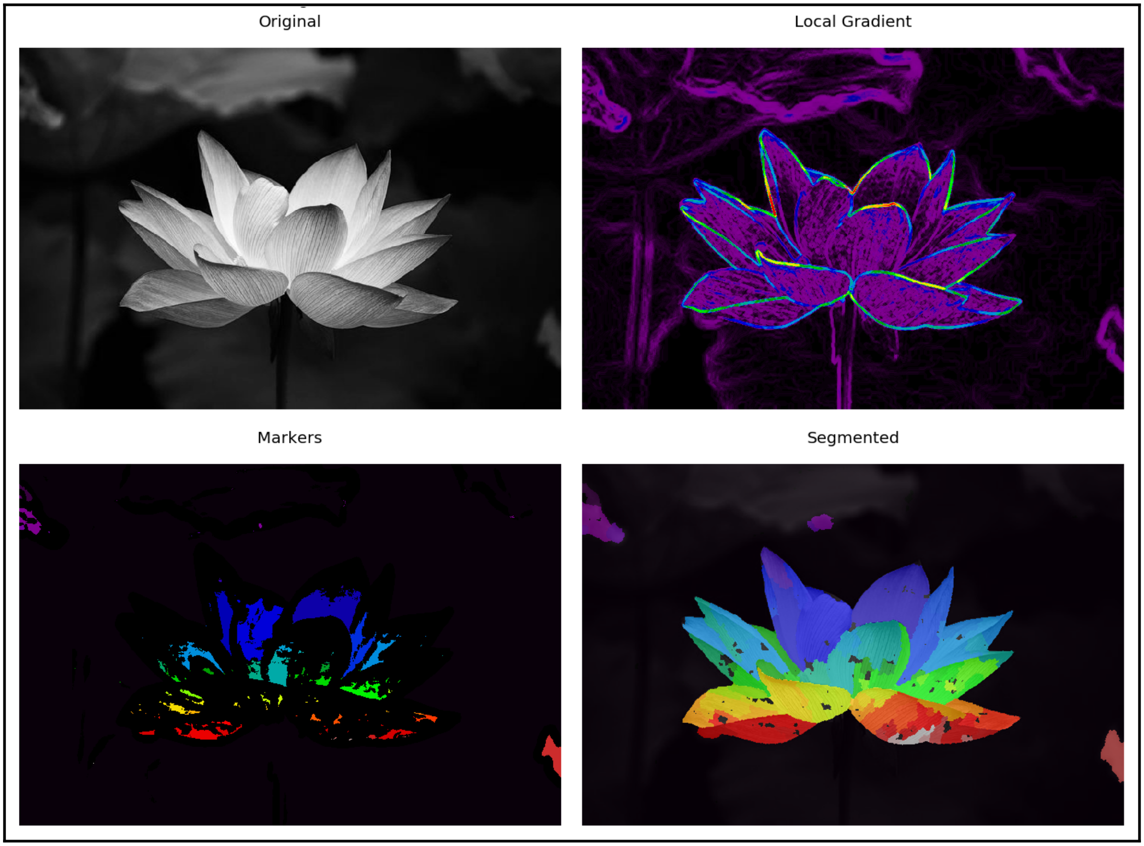

Structuring element 1
Structuring element 2

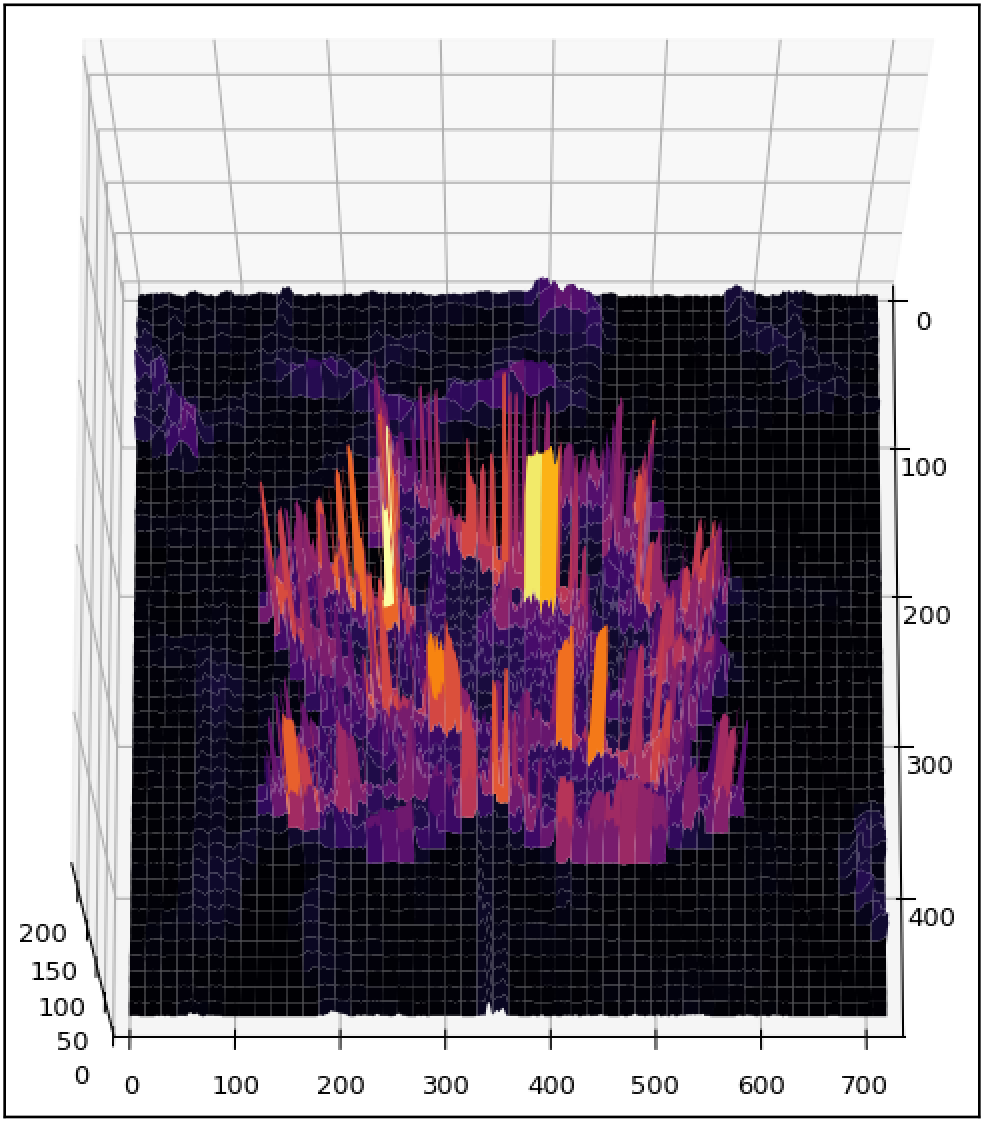

Original Image
Output with Hit-or-Miss Transform
ভারততীর্থ
হে মোর চিত্ত, পুণ্য তীর্থে জাগো রে ধীরে
এই ভারতের মহামানবের সাগরতীরে।
হেথায় দাঁড়ায়ে দু বাহু বাড়ায়ে নমি নরদেবতারে—
উদার ছন্দে, পরমানন্দে বন্দন করি তাঁরে।
ধ্যানগম্ভীর এই-যে ভূধর, নদী-জপমালা-ধৃত প্রান্তর,
হেথায় নিত্য হেরো পবিত্র ধরিত্রীরে—
এই ভারতের মহামানবের সাগরতীরে ॥
কেহ নাহি জানে কার আহ্বানে কত মানুষের ধারা
দুর্বার স্রোতে এল কোথা হতে, সমুদ্রে হল হারা।
হেথায় আর্য, হেথা অনার্য, হেথায় দ্রাবিড় চীন—
শক-হুন-দল পাঠান-মোগল এক দেহে হল লীন।
পশ্চিমে আজি খুলিয়াছে দ্বার, সেথা হতে সবে আনে উপহার,
দিবে আর নিবে, মিলাবে মিলিবে, যাবে না ফিরে—
এই ভারতের মহামানবের সাগরতীরে ॥
এসো হে আর্য, এসো অনার্য, হিন্দু-মুসলমান।
এসো এসো আজ তুমি ইংরাজ, এসো এসো খৃস্টান।
এসো ব্রাহ্মণ, শুচি করি মন ধরো হাত সবাকার।
এসো হে পতিত, হোক অপনীত সব অপমানভার।
মার অভিষেকে এসো এসো ত্বরা, মঙ্গলঘট হয় নি যে ভরা
সবার-পরশে-পবিত্র-করা তীর্থনীরে—
আজি ভারতের মহামানবের সাগরতীরে ॥

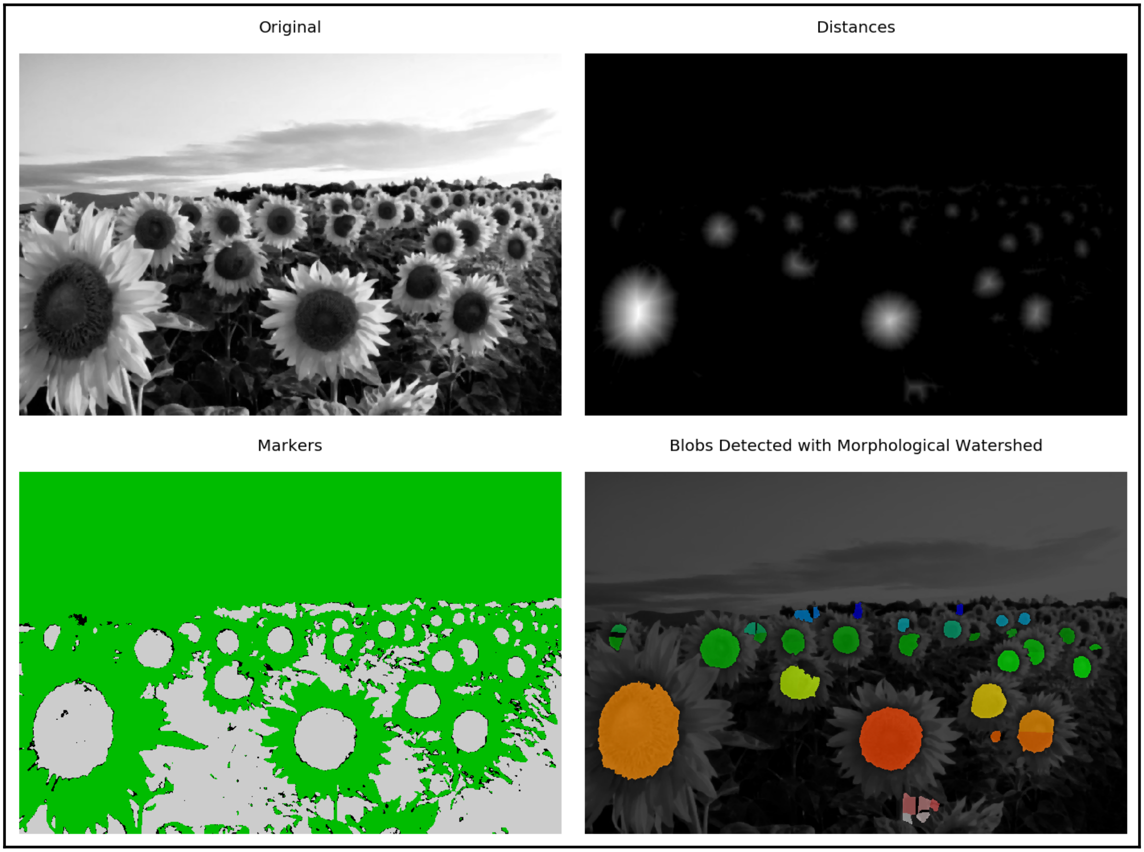
Finding -O- and =O in Triglyceride molecule



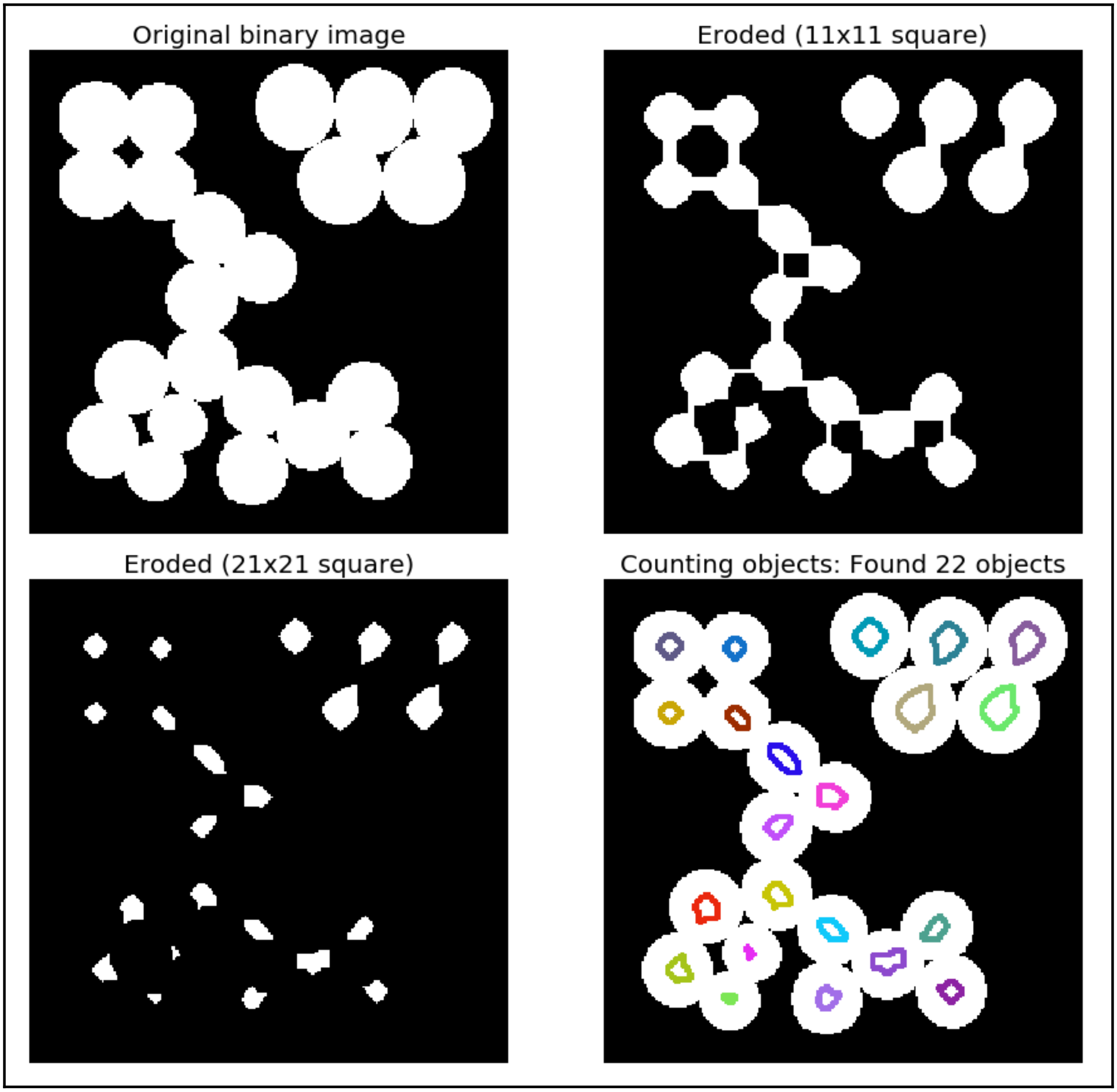

Original
Distance
Markers
Segmented


Original
Local Gradient
Markers
Segmented

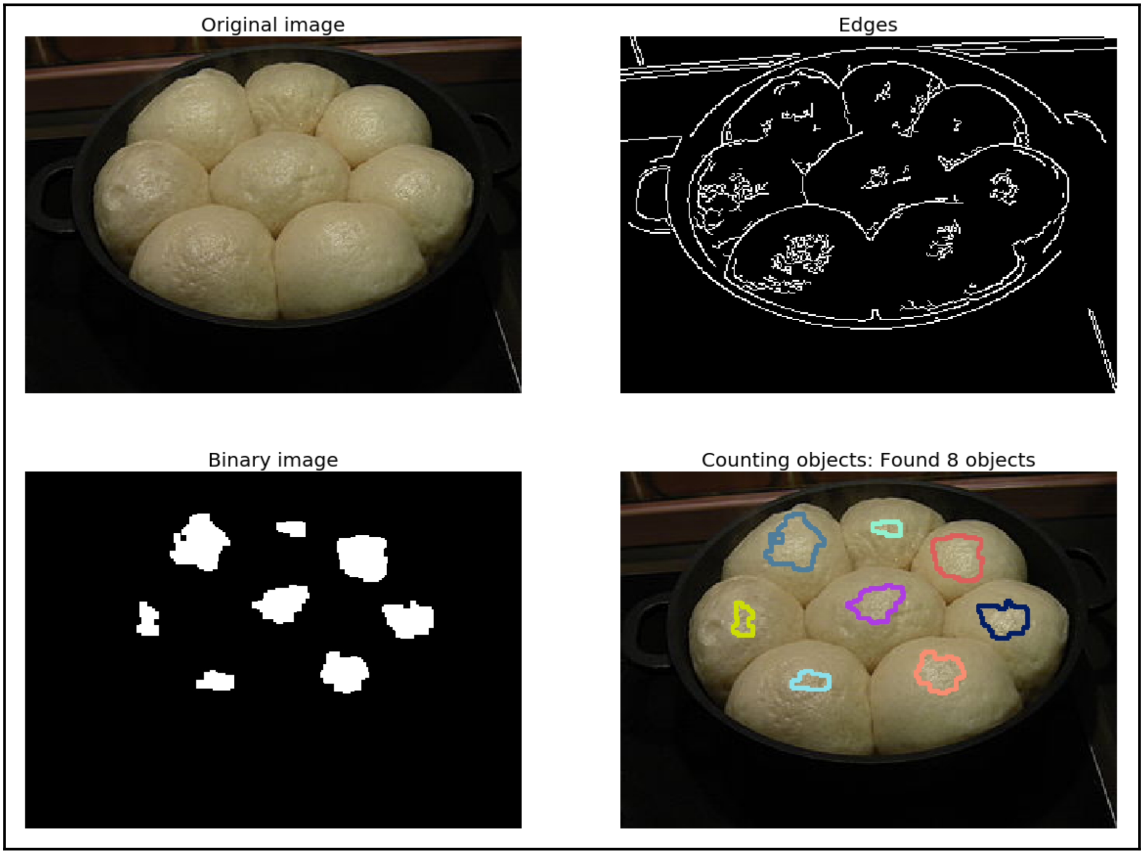

0
100
200
300
400
200
150
100
50
0
0
100
200
300
400
500
600
700

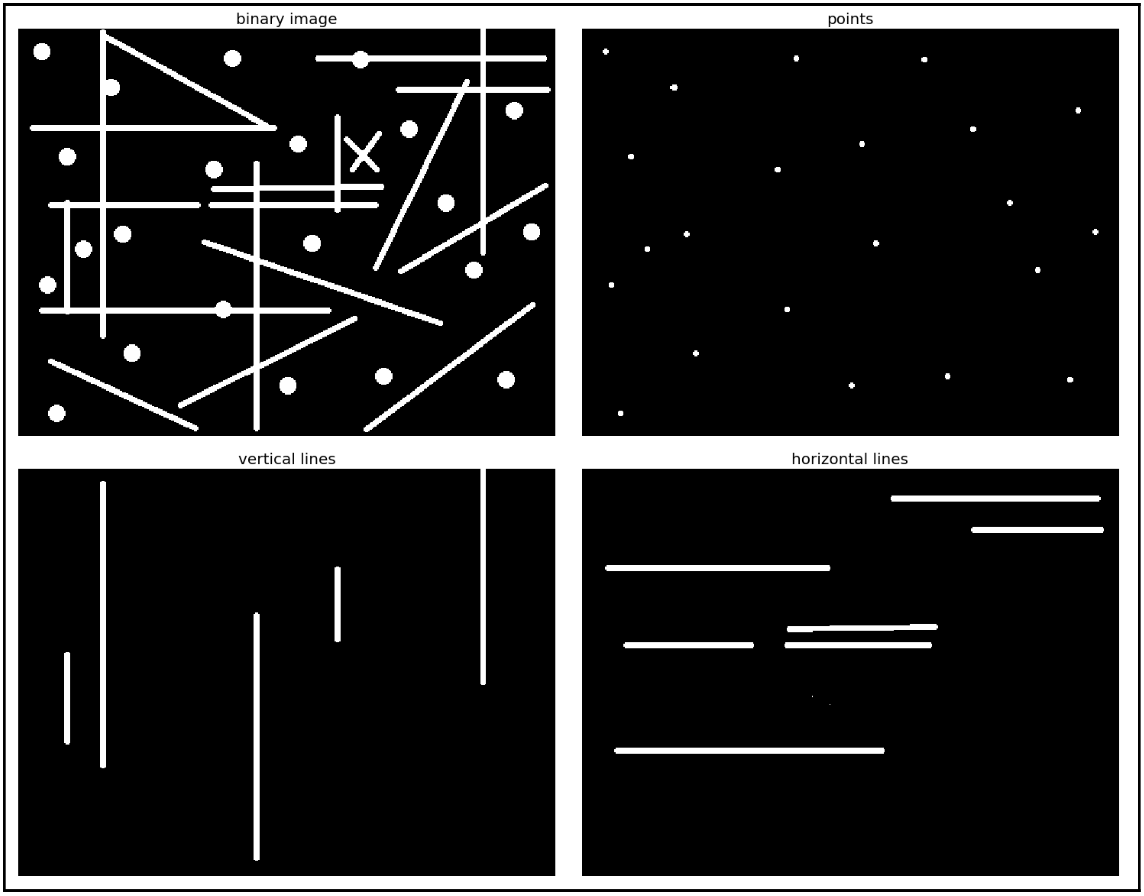

Original
Distances
Markers
Blobs Detected with Morphological Watershed

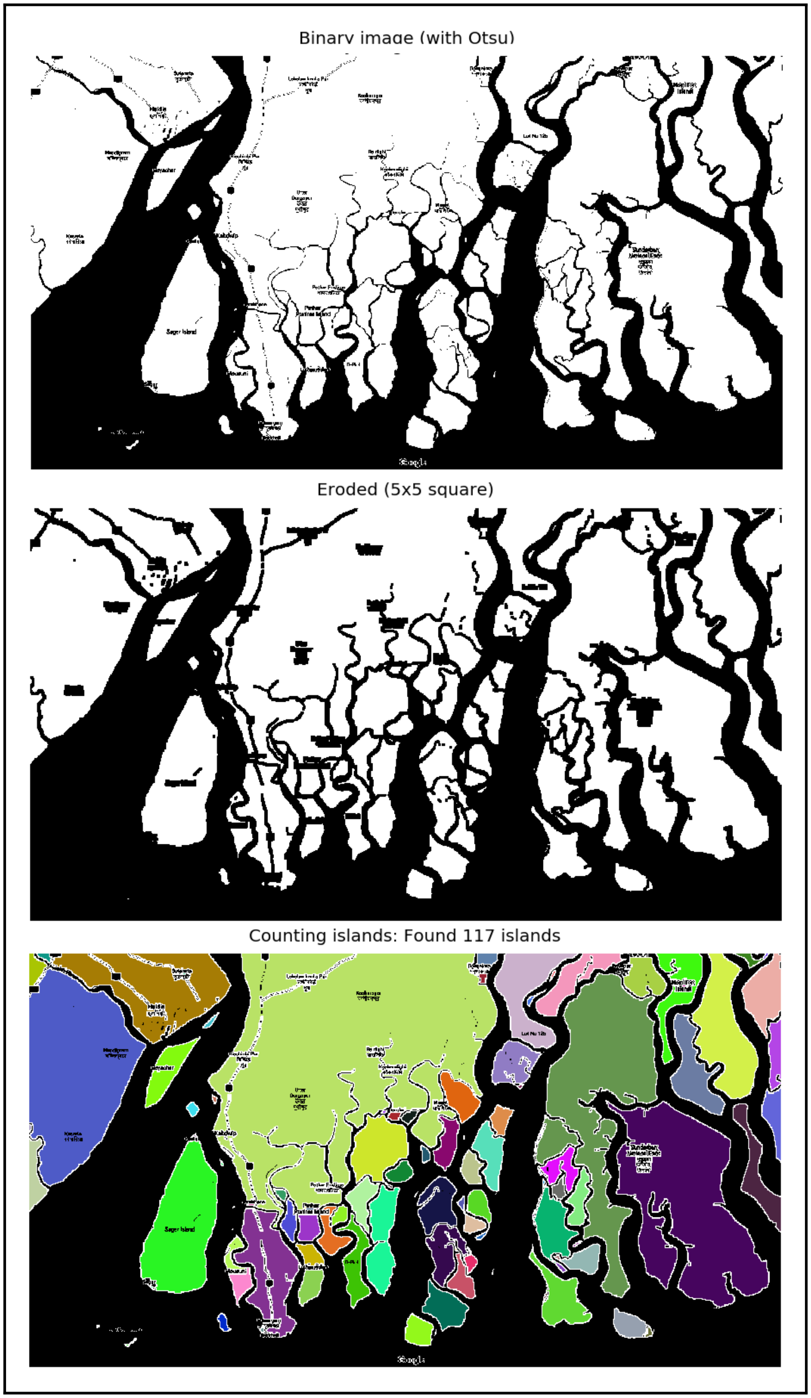

Original
Blobs Detected with LOG scale-space



Original binary image
Eroded (11x11 square)
Eroded (21x21 square)
Counting objects: Found 22 objects


Original image
Edges
Binary image
Counting objects: Found 8 objects


binary image
points
vertical lines
horizontal lines


Binary image (with Otsu)
Eroded (5x5 square)
Counting islands: Found 117 islands

Chapter 5: Image Registration

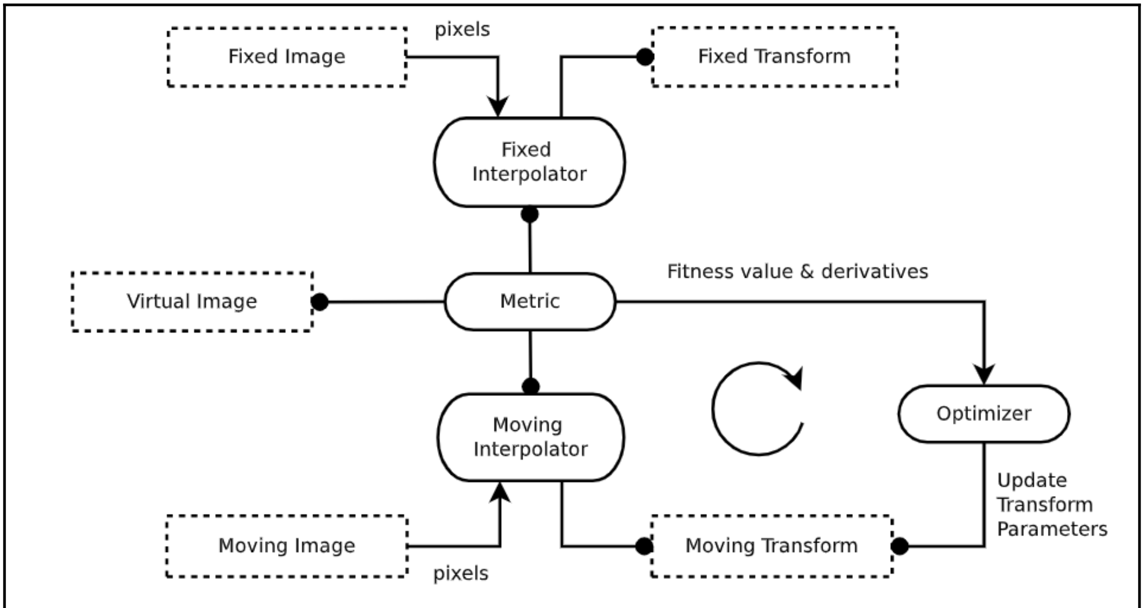
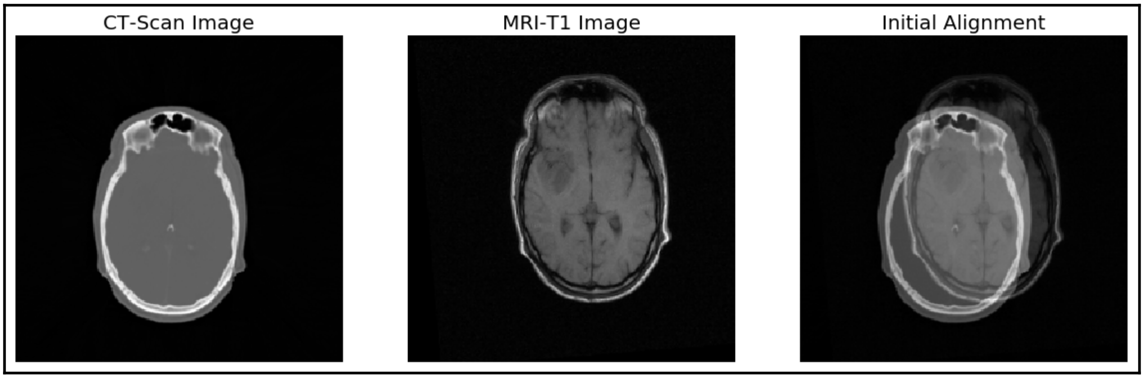
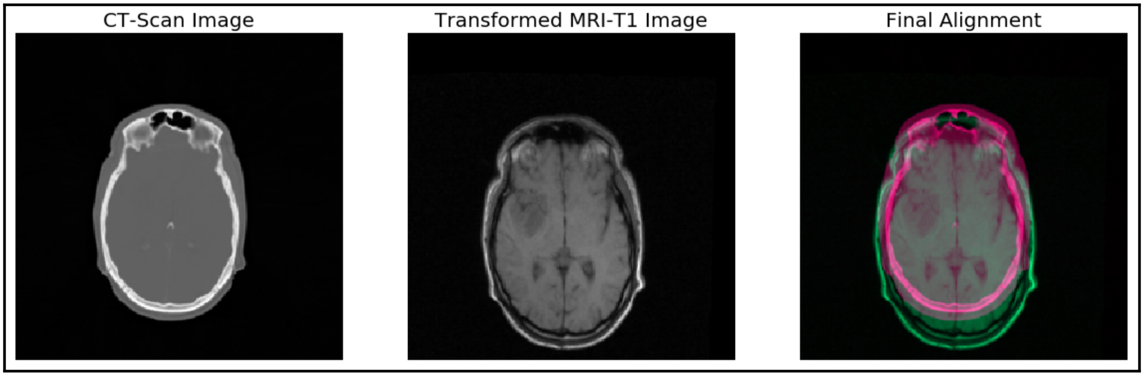
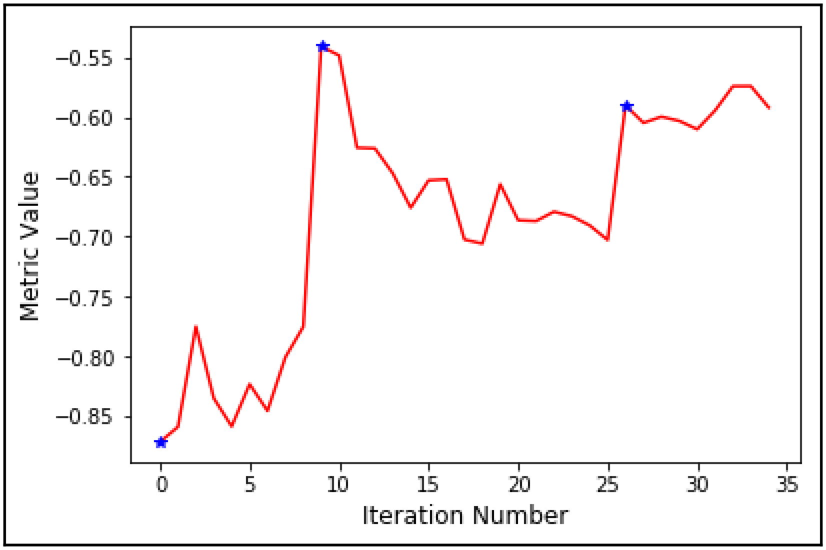
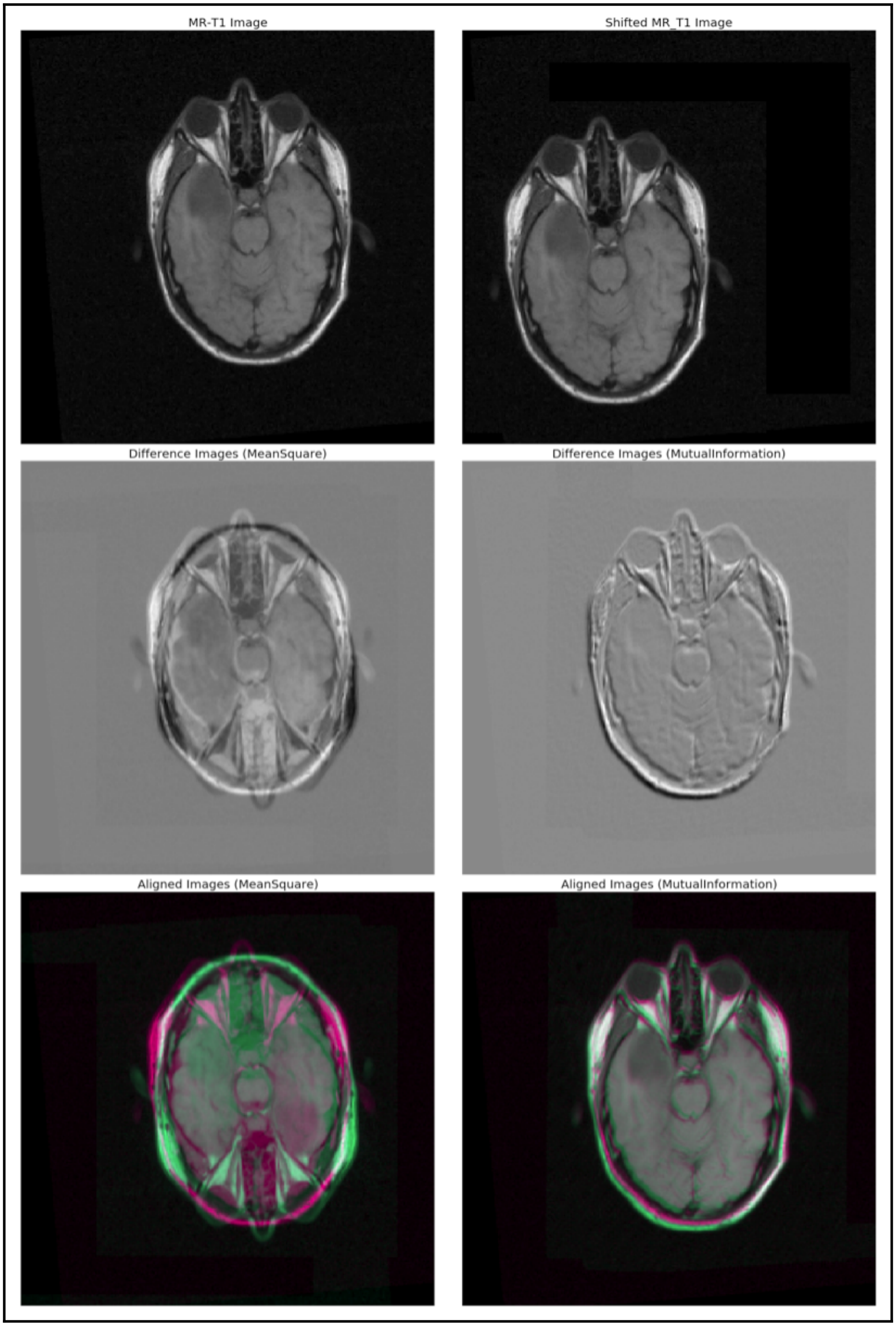


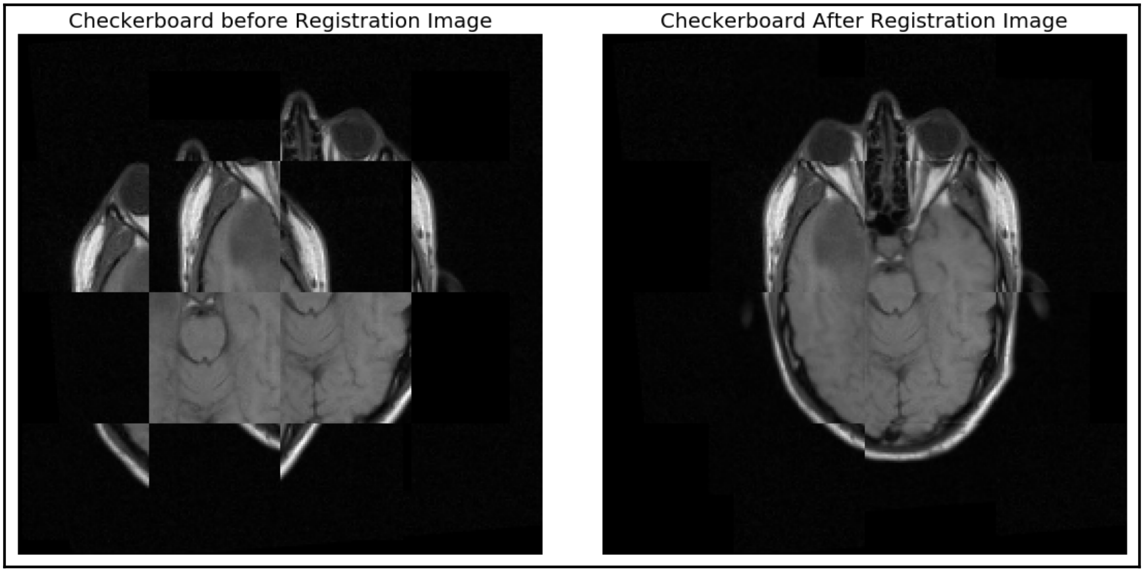
image from *http://insightsoftwareconsortium.github.io/SimpleITK-Notebooks/Python_html/60_Registration_Introduction.html*




CT-Scan Image
Transformed MRI-T1 Image
Final Alignment


−0.55
−0.60
−0.65
−0.70
−0.75
−0.80
−0.85
Metric Value
0
5
10
15
20
25
30
35
Iteration Number


MR-T1 Image
Shifted MR_T1 Image
Difference Images (MeanSquare)
Difference Images (MutualInformation)
Aligned Images (MeanSquare)
Aligned Images (MutualInformation)



Estimate p: $I_r(\mathbf{x}) = I_w(\phi(\mathbf{x};\mathbf{p})), \quad \forall\, \mathbf{x} \in \mathcal{T}$

Optimization problem: $\min_{\mathbf{p},\boldsymbol{\alpha}} E(\mathbf{p},\boldsymbol{\alpha}) = \min_{\mathbf{p},\boldsymbol{\alpha}} \sum_{\mathbf{x}\in\mathcal{T}} |I_r(\mathbf{x}) - \Psi(I_w(\phi(\mathbf{x};\mathbf{p})),\boldsymbol{\alpha})|^p$

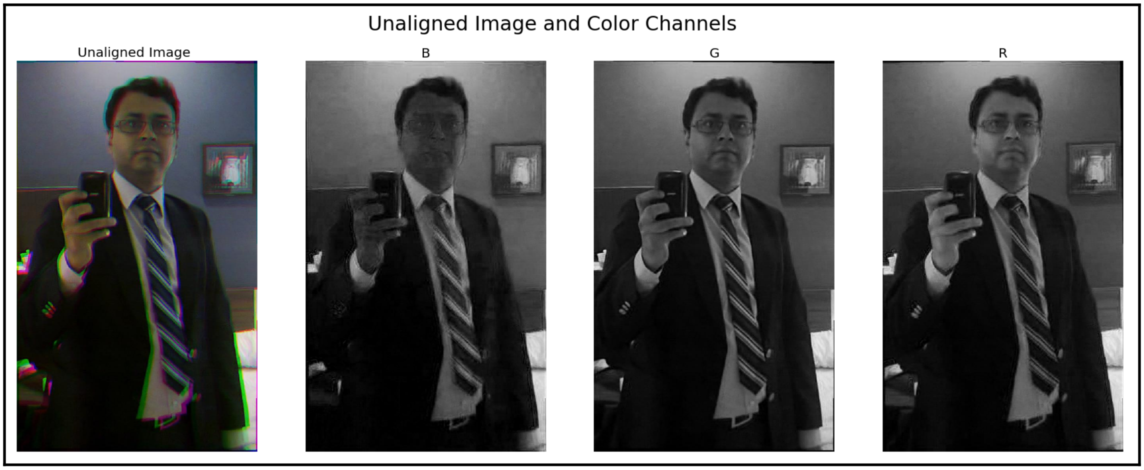
reference vector: $\mathbf{i}_r = [I_r(\mathbf{x}_1)\; I_r(\mathbf{x}_2) \cdots I_r(\mathbf{x}_K)]^t$

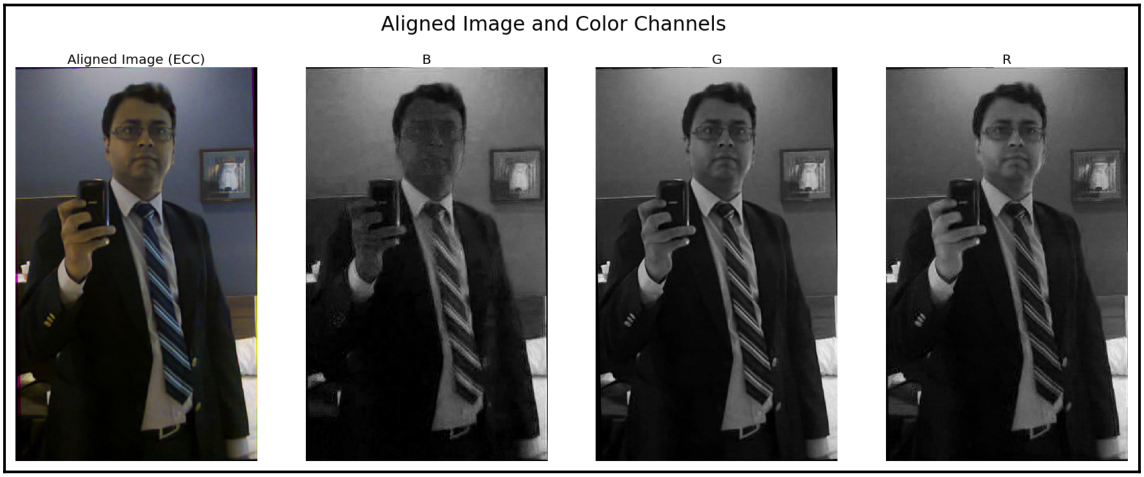
warped vector: $\mathbf{i}_w(\mathbf{p}) = [I_w(\mathbf{y}_1(\mathbf{p}))\; I_w(\mathbf{y}_2(\mathbf{p})) \cdots I_w(\mathbf{y}_K(\mathbf{p}))]^t$

(minimize) ECC criterion: $E_{\mathrm{ECC}}(\mathbf{p}) = \left\| \frac{\bar{\mathbf{i}}_r}{\|\bar{\mathbf{i}}_r\|} - \frac{\bar{\mathbf{i}}_w(\mathbf{p})}{\|\bar{\mathbf{i}}_w(\mathbf{p})\|} \right\|^2$

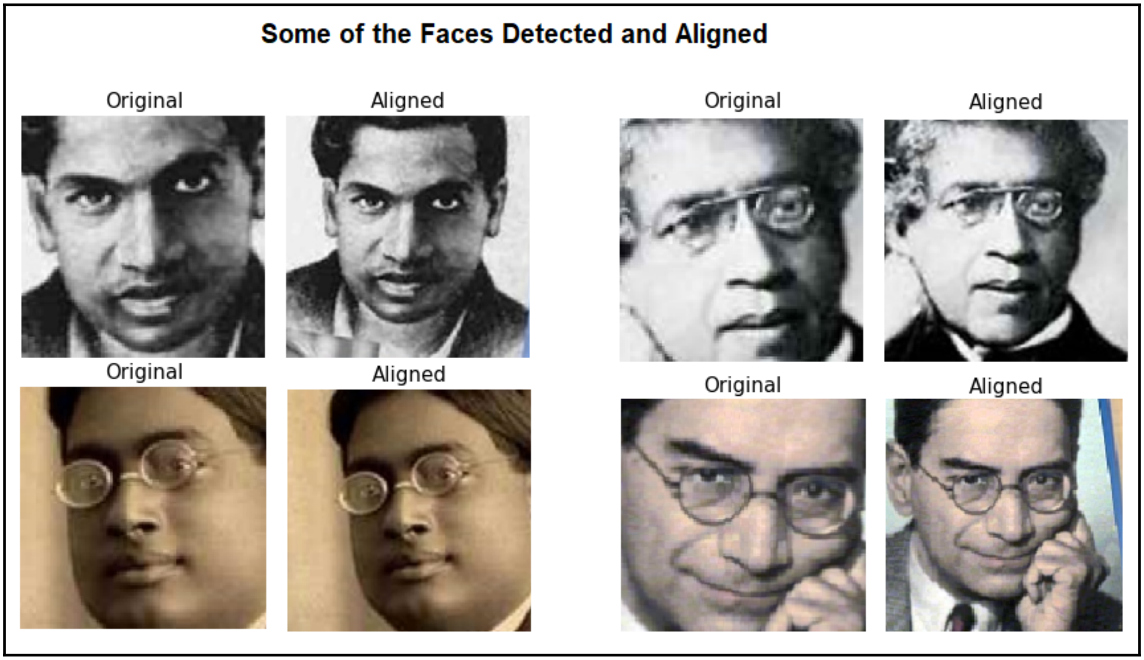
(maximize) enhanced correlation coefficient: $\rho(\mathbf{p}) = \frac{\bar{\mathbf{i}}_r^t \bar{\mathbf{i}}_w(\mathbf{p})}{\|\bar{\mathbf{i}}_r\| \|\bar{\mathbf{i}}_w(\mathbf{p})\|} = \hat{\mathbf{i}}_r^t \frac{\bar{\mathbf{i}}_w(\mathbf{p})}{\|\bar{\mathbf{i}}_w(\mathbf{p})\|}$

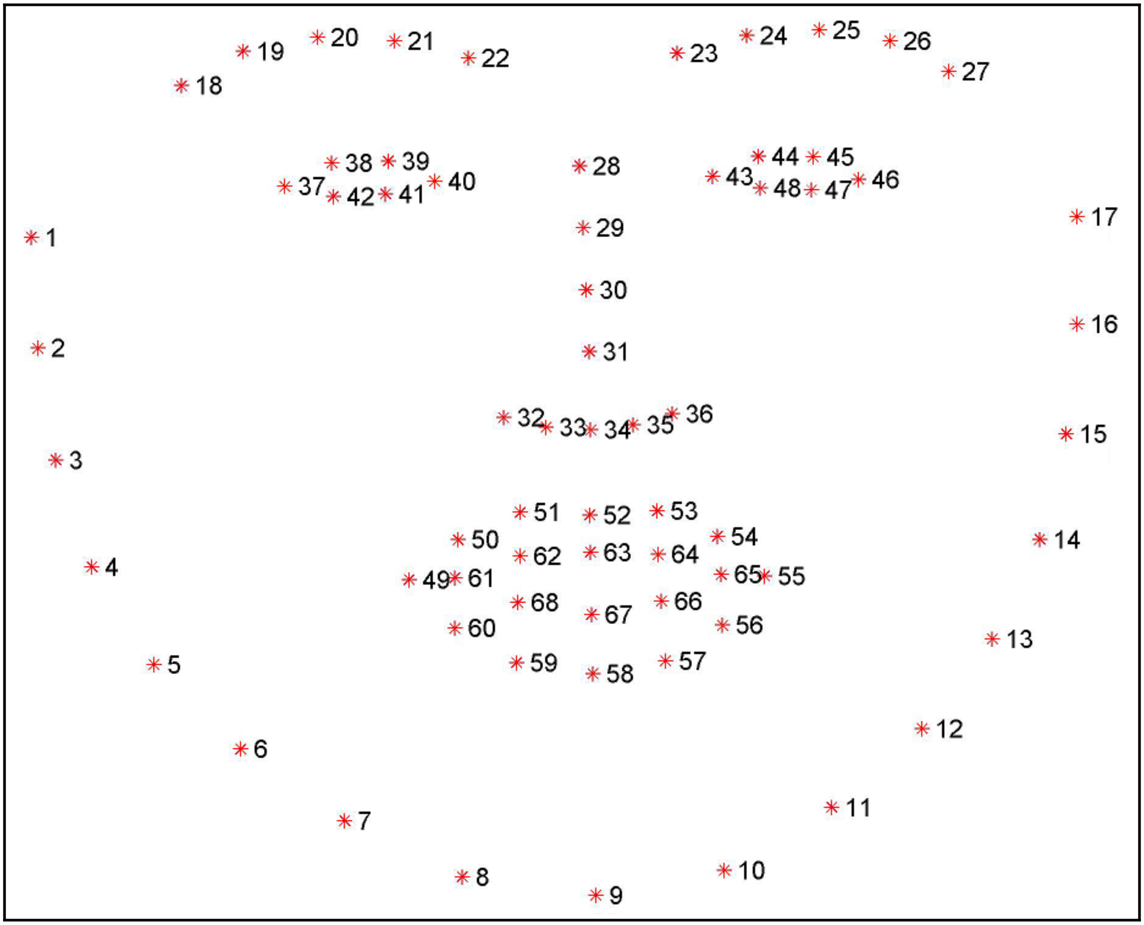
Taken from *http://xanthippi.ceid.upatras.gr/people/evangelidis/george_files/PAMI_2008.pdf*


Unaligned Image and Color Channels
Unaligned Image
B
G
R

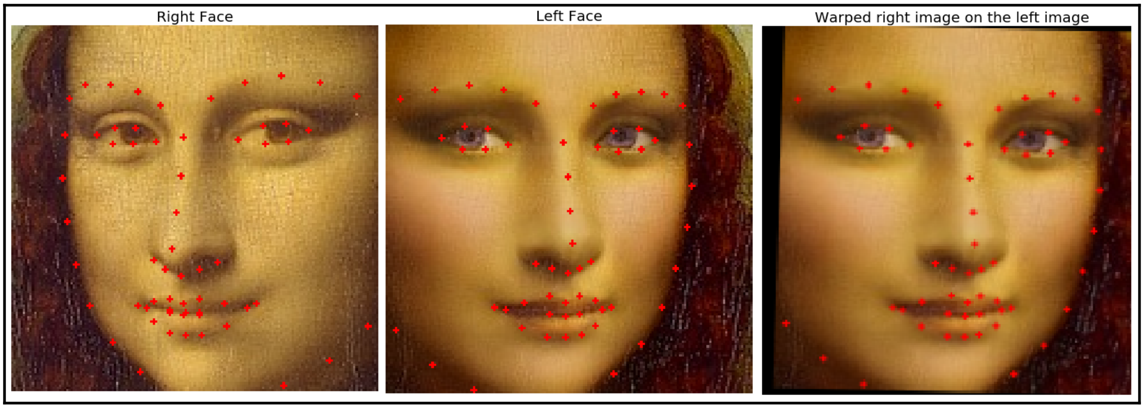

Aligned Image and Color Channels
Aligned Image (ECC)
B
G
R

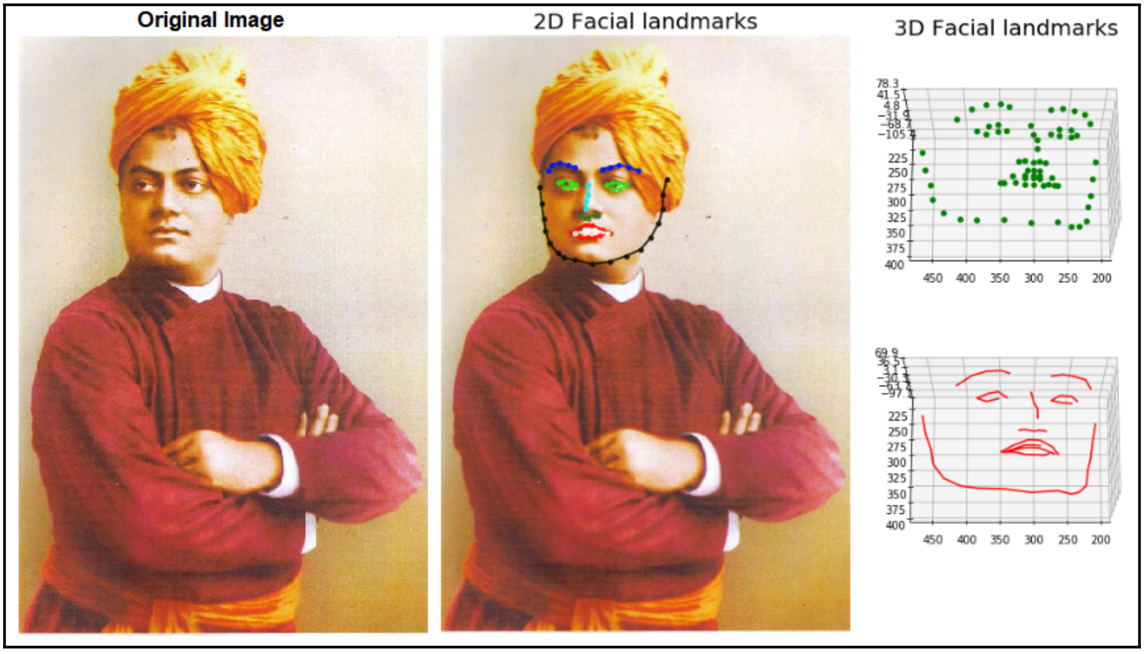
Original Image: Famous Indian Scientists


Some of the Faces Detected and Aligned
Original
Aligned
Original
Aligned
Original
Aligned
Original
Aligned



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
18 19 20 21 22 23 24 25 26 27
28 29 30 31 32 33 34 35 36
37 38 39 40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68


Right Face
Left Face
Warped right image on the left image



RANSAC Algorithm

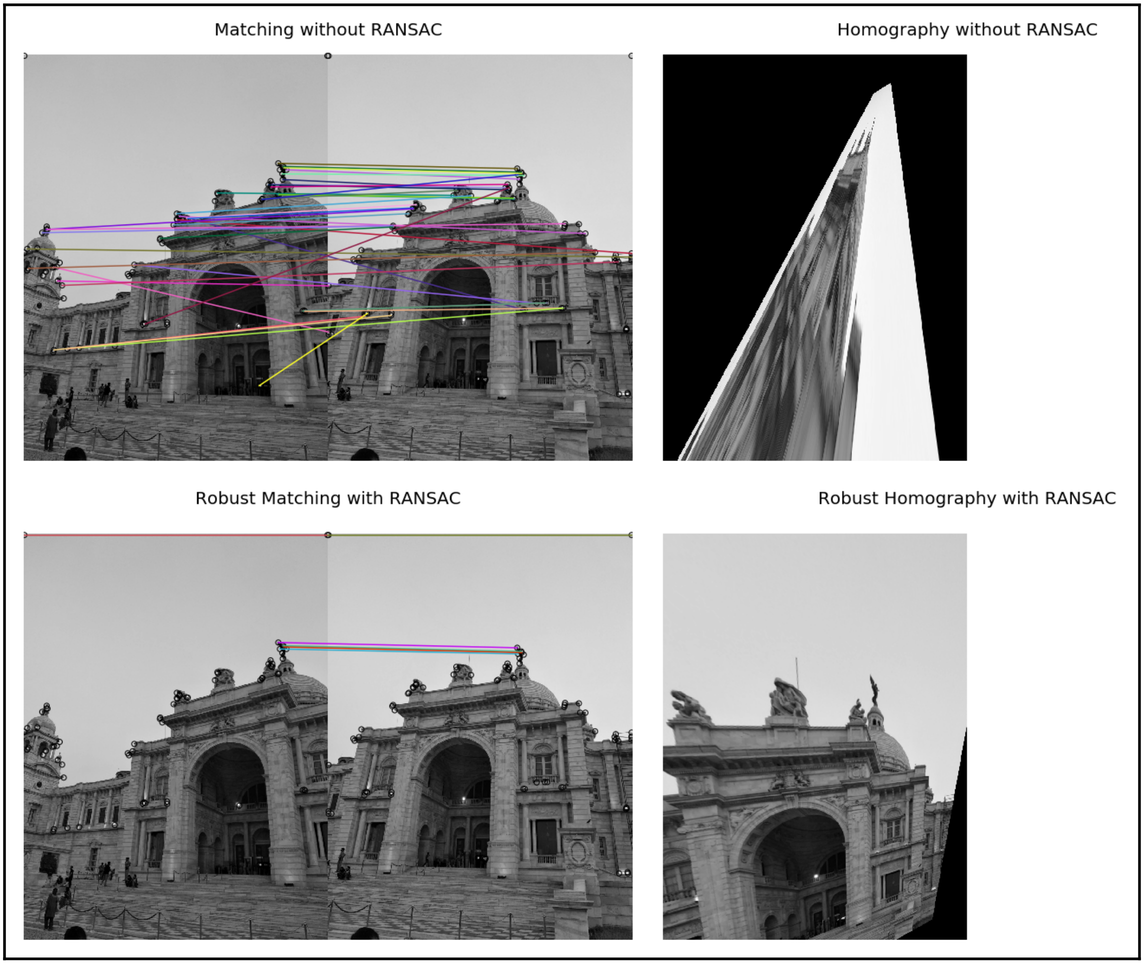
$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}$$

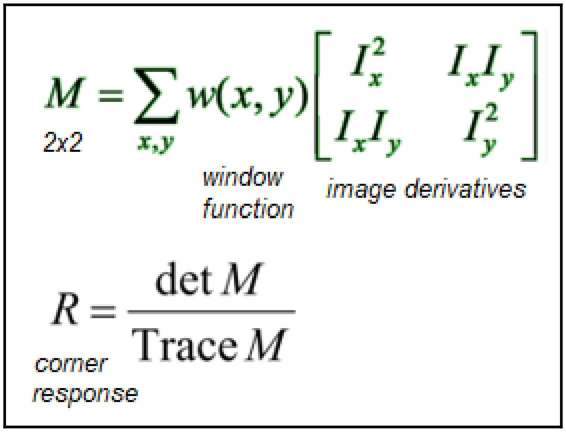
probability that there is at least one sample free from outliers

number of trials needed

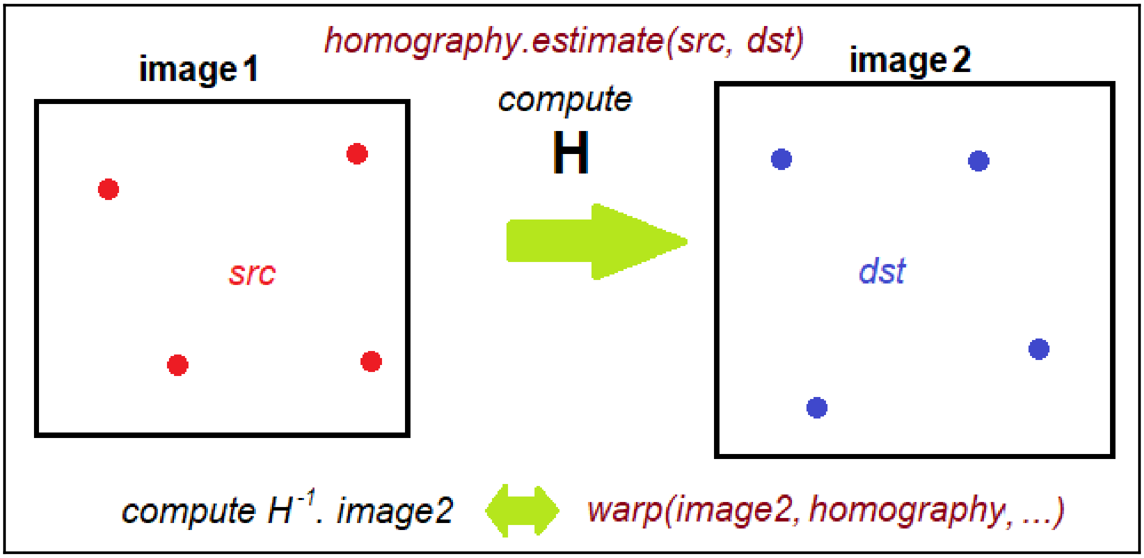
probabilty of an outlier

sample size (= 4 for 8-point algo)

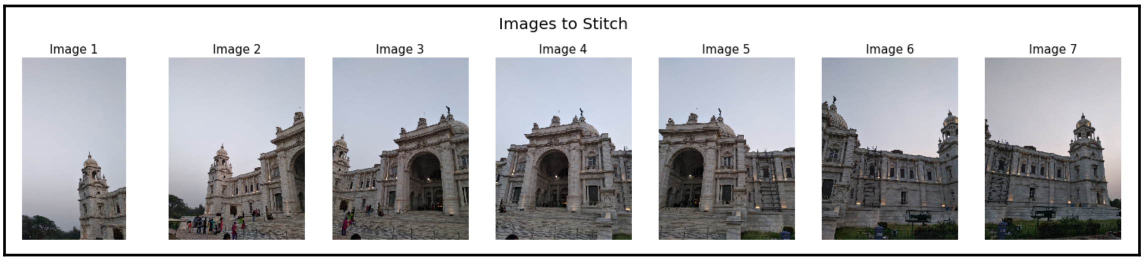

Matching without RANSAC
Homography without RANSAC
Robust Matching with RANSAC
Robust Homography with RANSAC



$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$
2x2
window function
image derivatives
$R = \frac{\det M}{\mathrm{Trace}\, M}$
corner response



homography.estimate(src, dst)
image 1
image 2
compute
H
src
dst
compute H^{-1}. image2
warp(image2, homography, ...)



Images to Stitch
Image 1
Image 2
Image 3
Image 4
Image 5
Image 6
Image 7

Final Panorama Image

Final Panorama Image

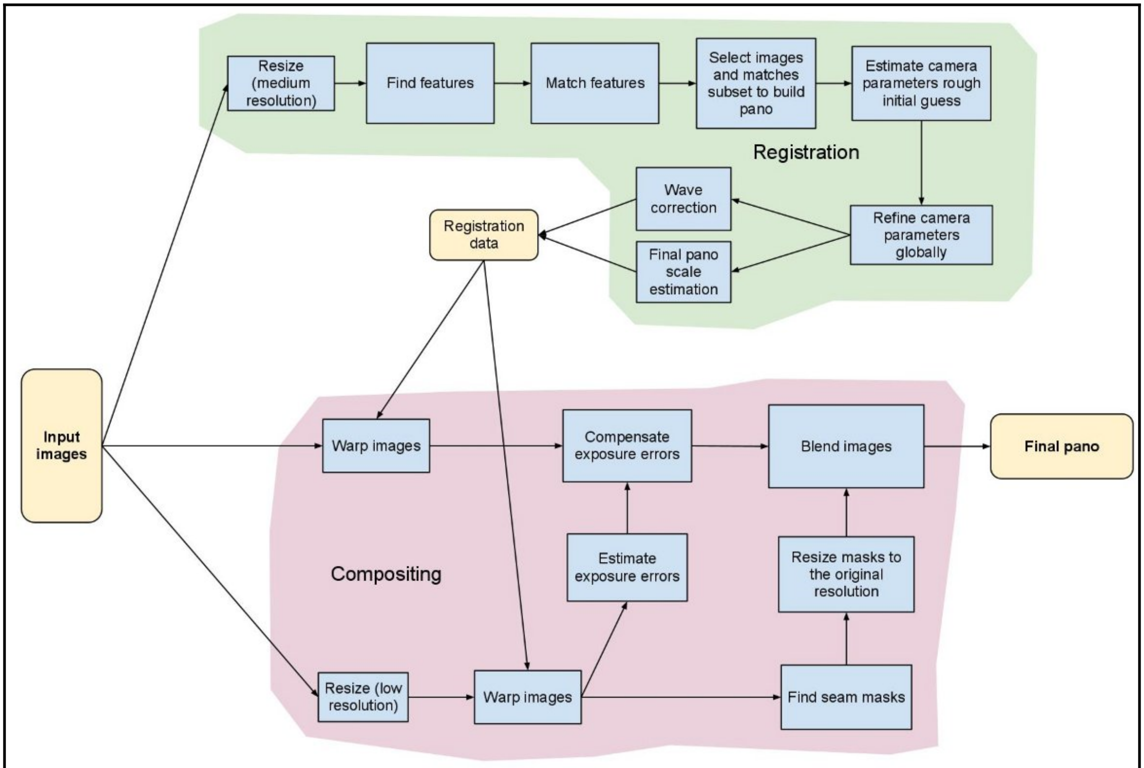
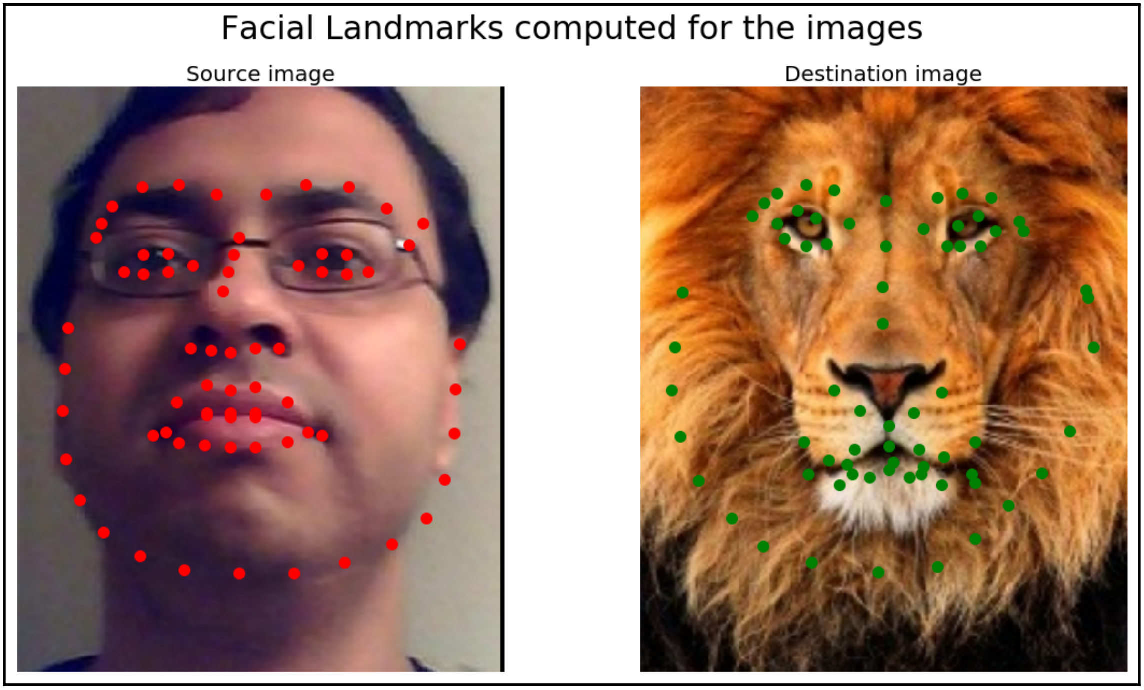
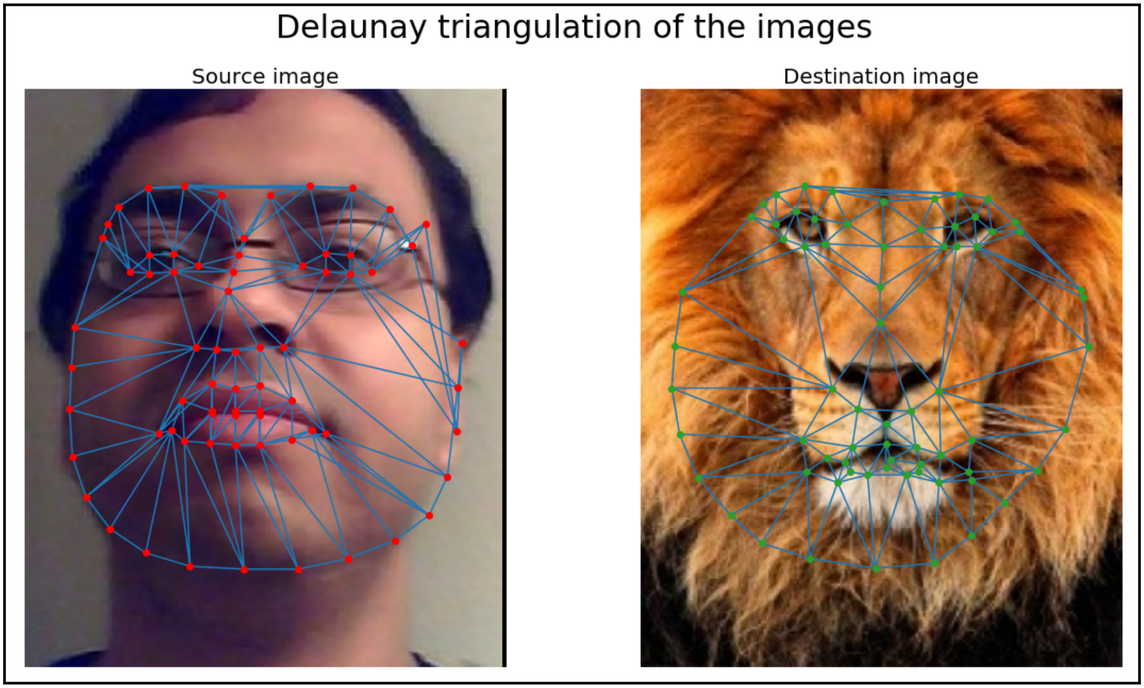
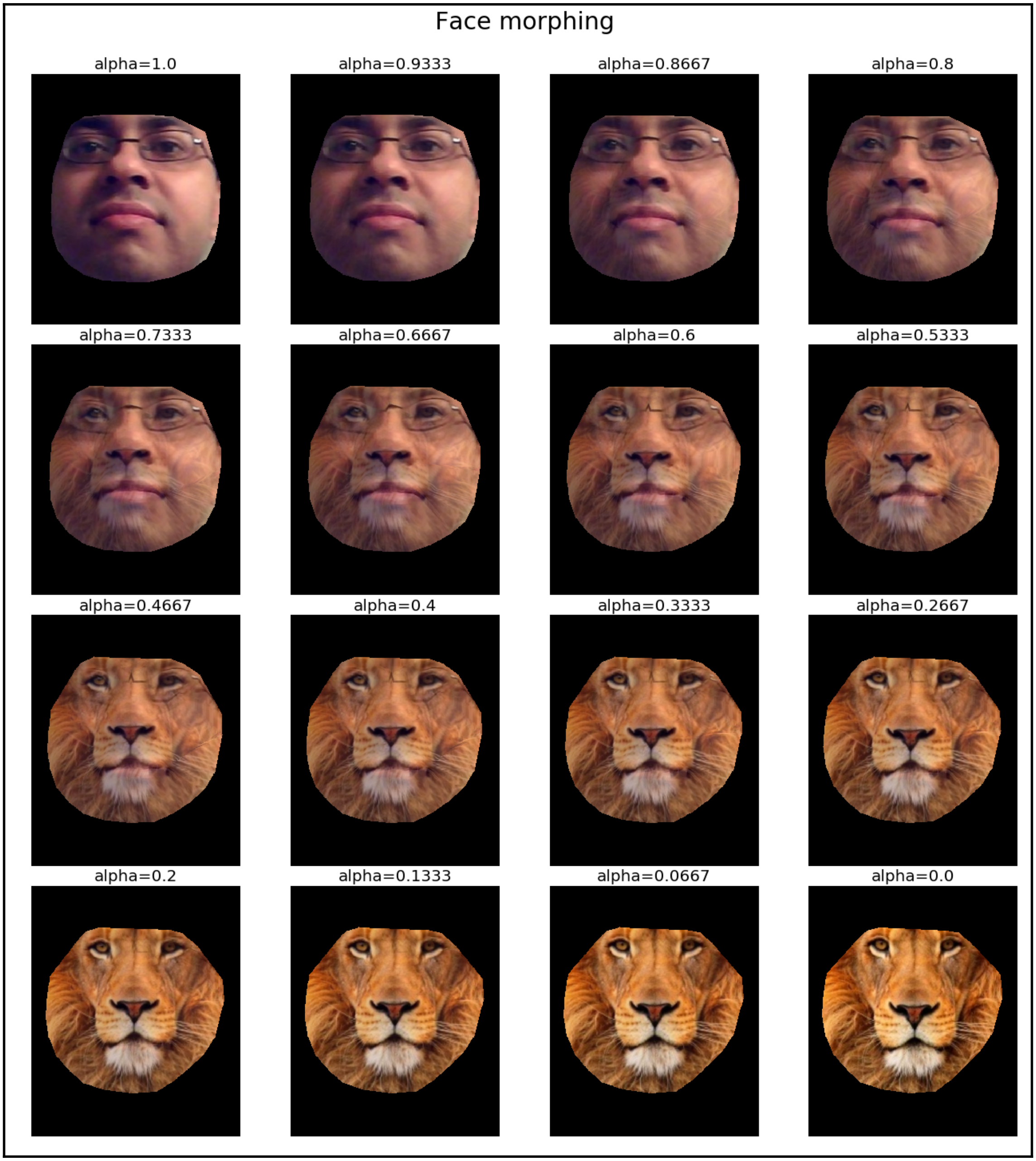
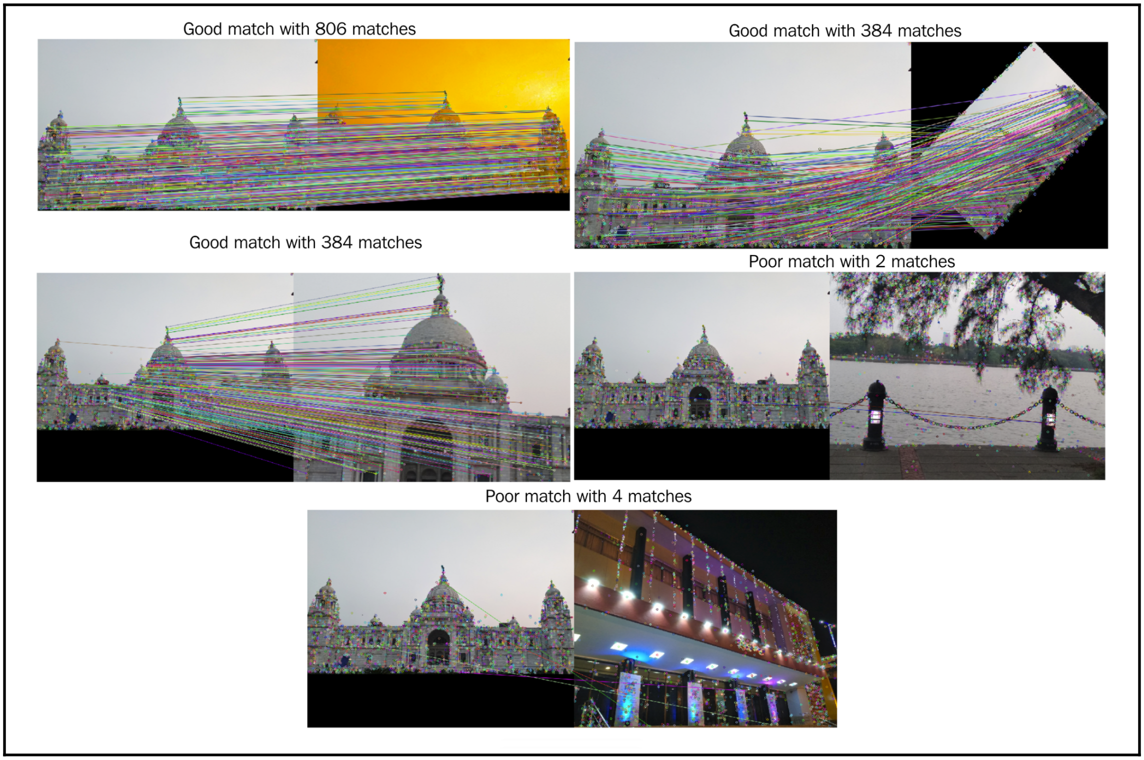


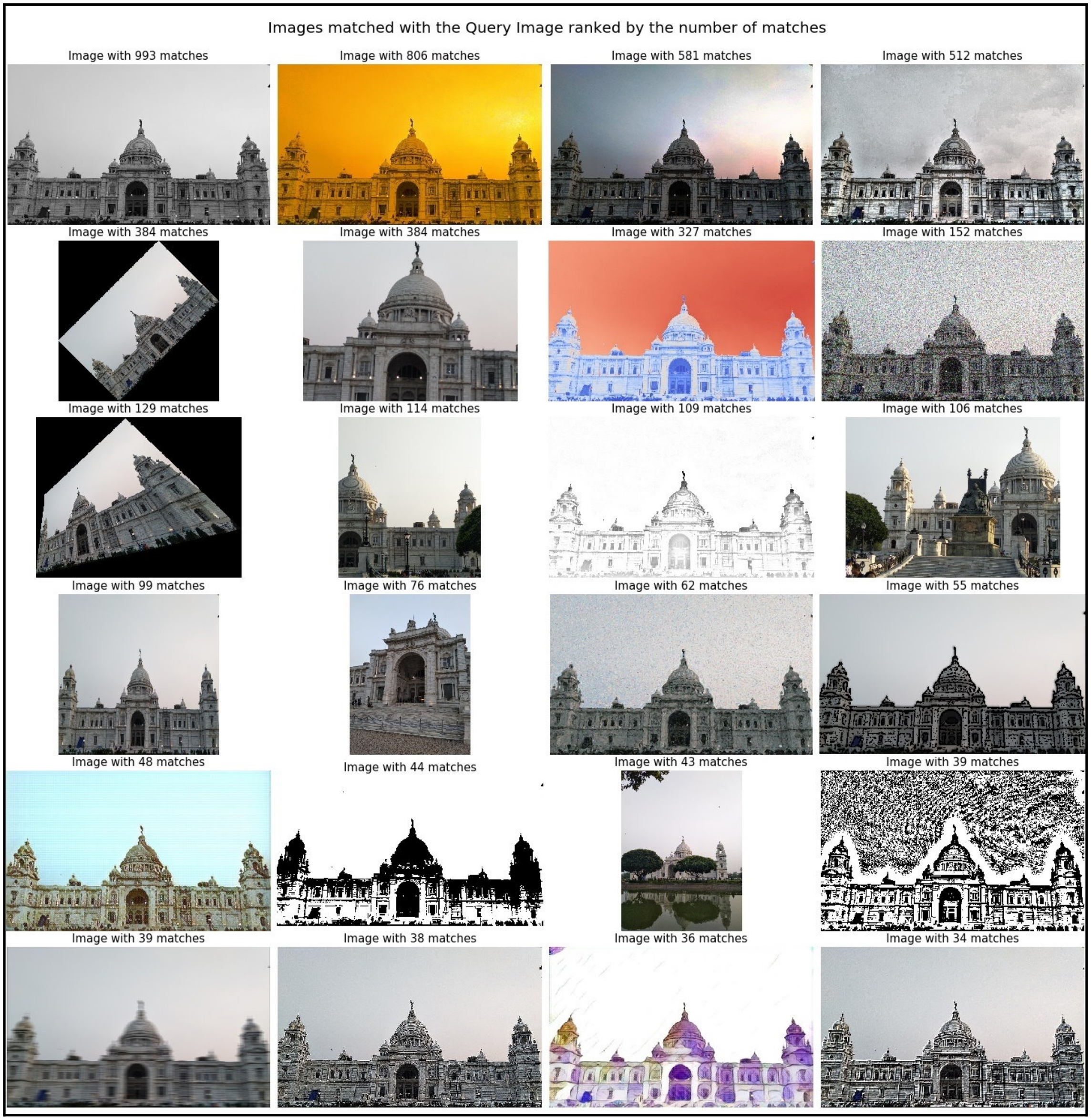
Figure taken from https://docs.opencv.org/3.4.7/d1/d46/group__stitching.html

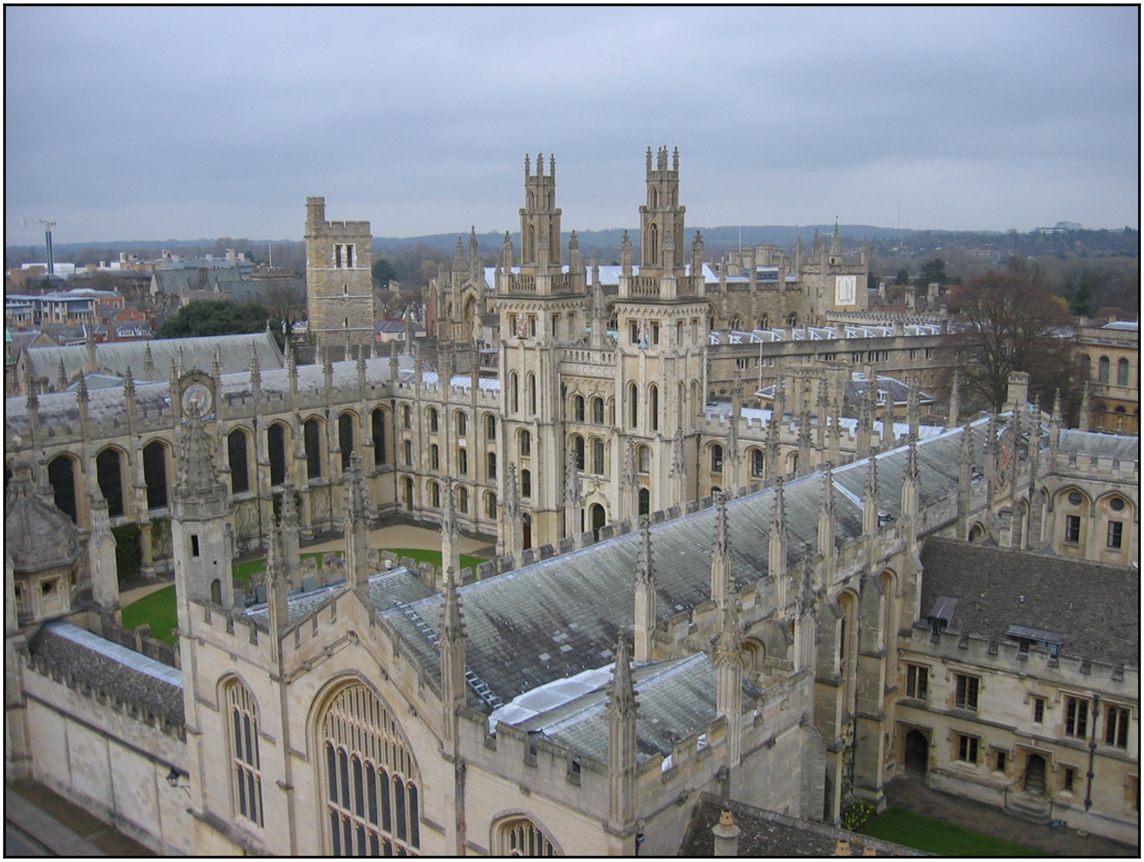

Facial Landmarks computed for the images
Source image
Destination image


Delaunay triangulation of the images
Source image
Destination image

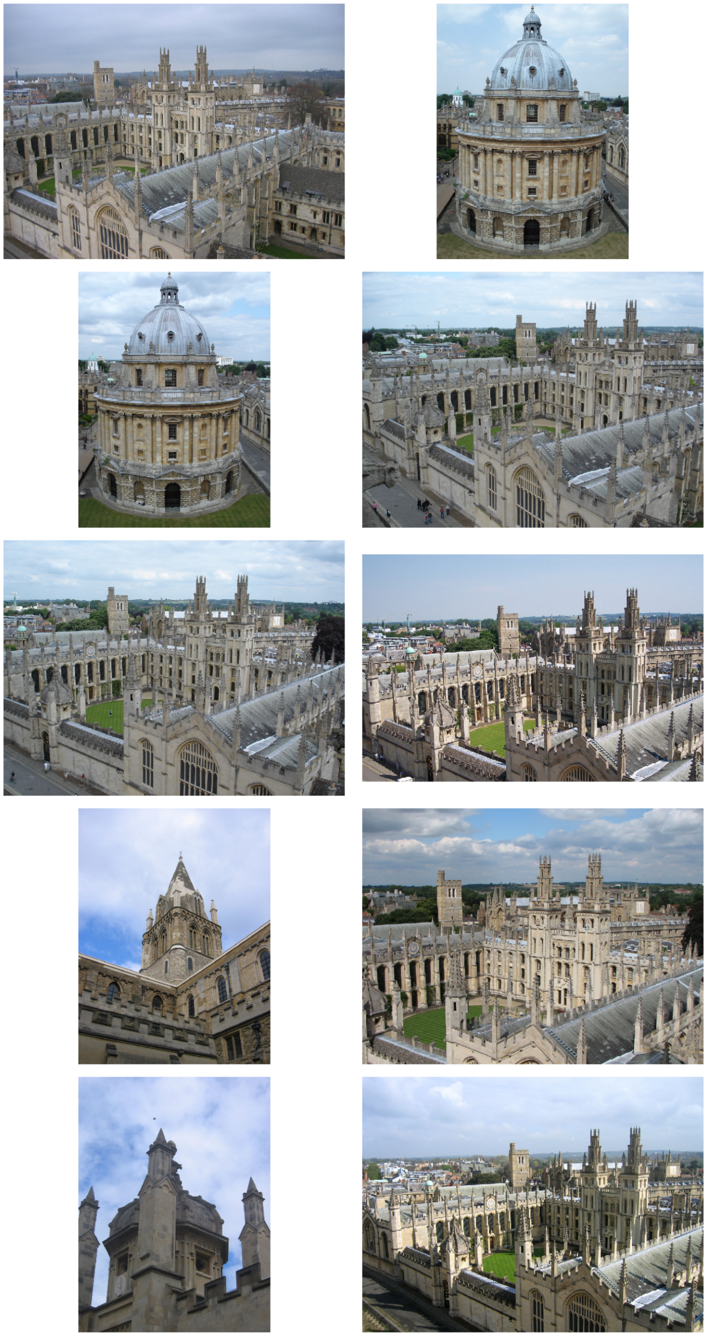

Face morphing
alpha=1.0
alpha=0.9333
alpha=0.8667
alpha=0.8
alpha=0.7333
alpha=0.6667
alpha=0.6
alpha=0.5333
alpha=0.4667
alpha=0.4
alpha=0.3333
alpha=0.2667
alpha=0.2
alpha=0.1333
alpha=0.0667
alpha=0.0


Good match with 806 matches
Good match with 384 matches
Good match with 384 matches
Poor match with 2 matches
Poor match with 4 matches


Images matched with the Query Image ranked by the number of matches
Image with 993 matches
Image with 806 matches
Image with 581 matches
Image with 512 matches
Image with 384 matches
Image with 384 matches
Image with 327 matches
Image with 152 matches
Image with 129 matches
Image with 114 matches
Image with 109 matches
Image with 106 matches
Image with 99 matches
Image with 76 matches
Image with 62 matches
Image with 55 matches
Image with 48 matches
Image with 44 matches
Image with 43 matches
Image with 39 matches
Image with 39 matches
Image with 38 matches
Image with 36 matches
Image with 34 matches



Top matched Images found by the Search Engine (with color histogram features)



Otsu Thresholding

$$q_1(t) = \sum_{i=0}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{L-1} P(i)$$

$$\mu_1(t) = \sum_{i=0}^{t} \frac{iP(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{L-1} \frac{iP(i)}{q_2(t)}$$

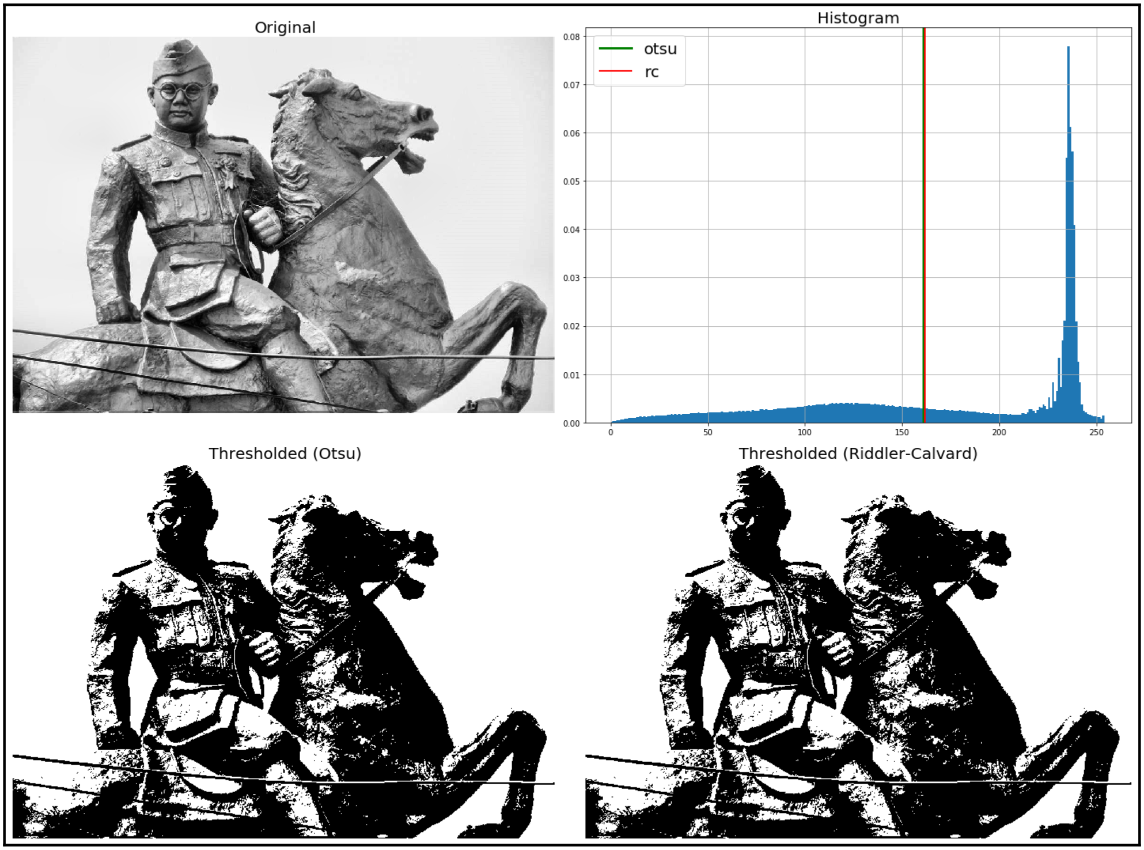
$$\sigma_1^2(t) = \sum_{i=0}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \qquad \sigma_2^2(t) = \sum_{i=t+1}^{L-1} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)}$$

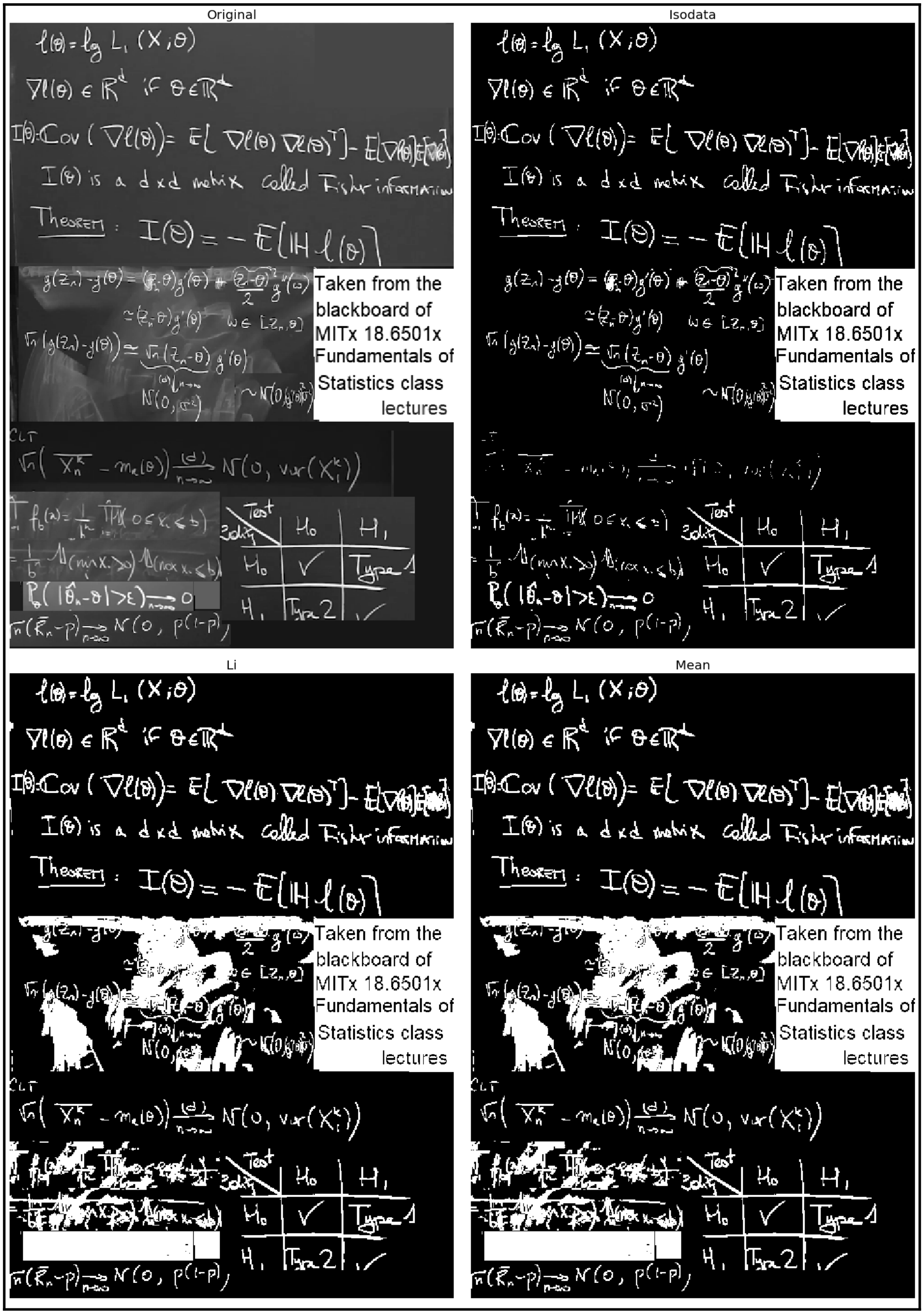
variance of class with pixel values $\leq$ **t**

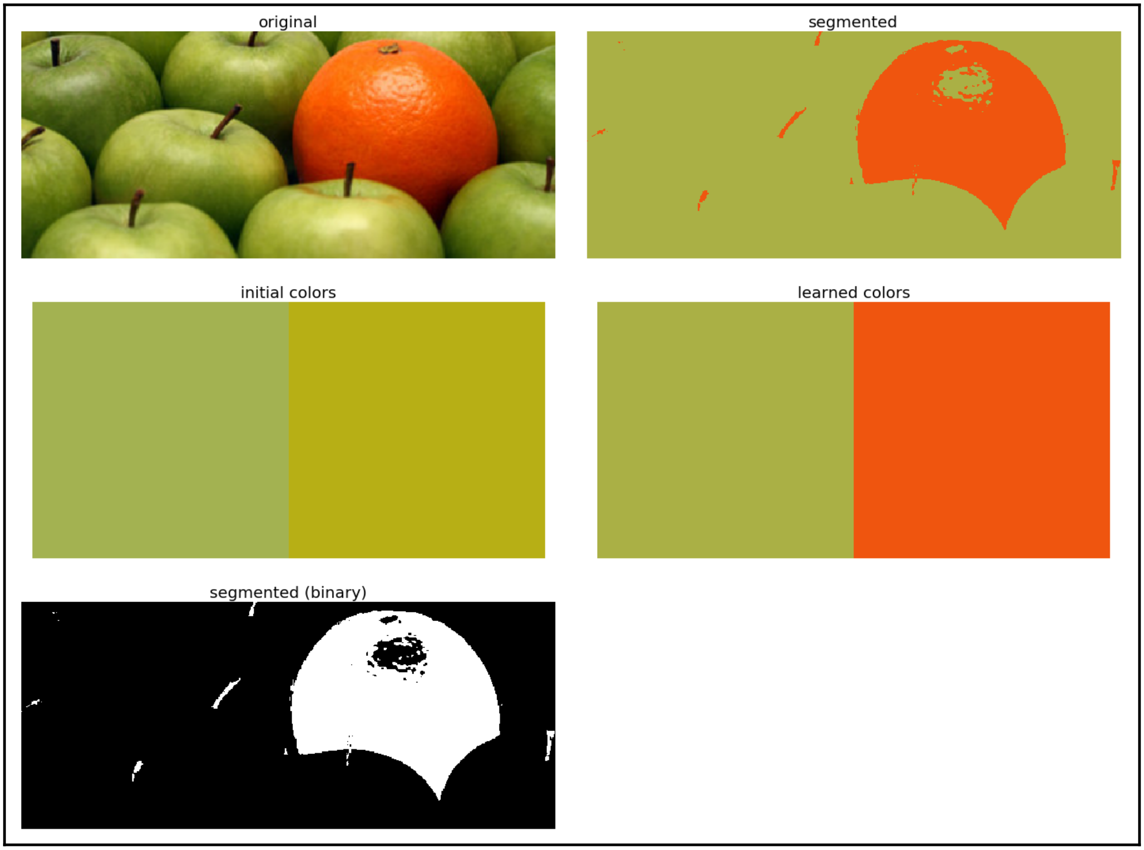
variance of class with pixel values **> t**

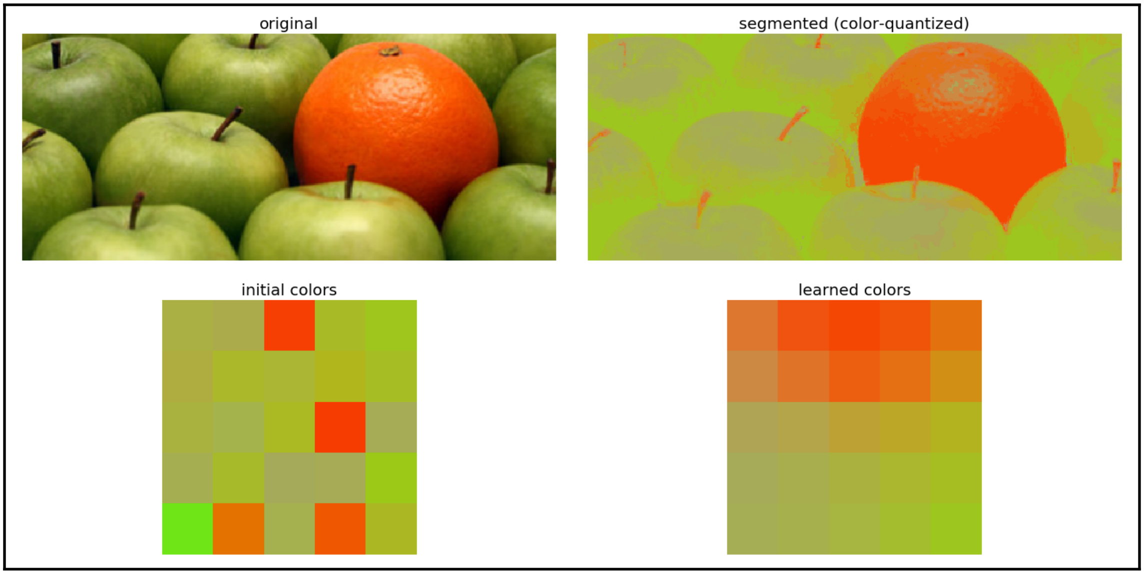
$$\underbrace{\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t)}_{\textit{Weighted within-class variance}}$$

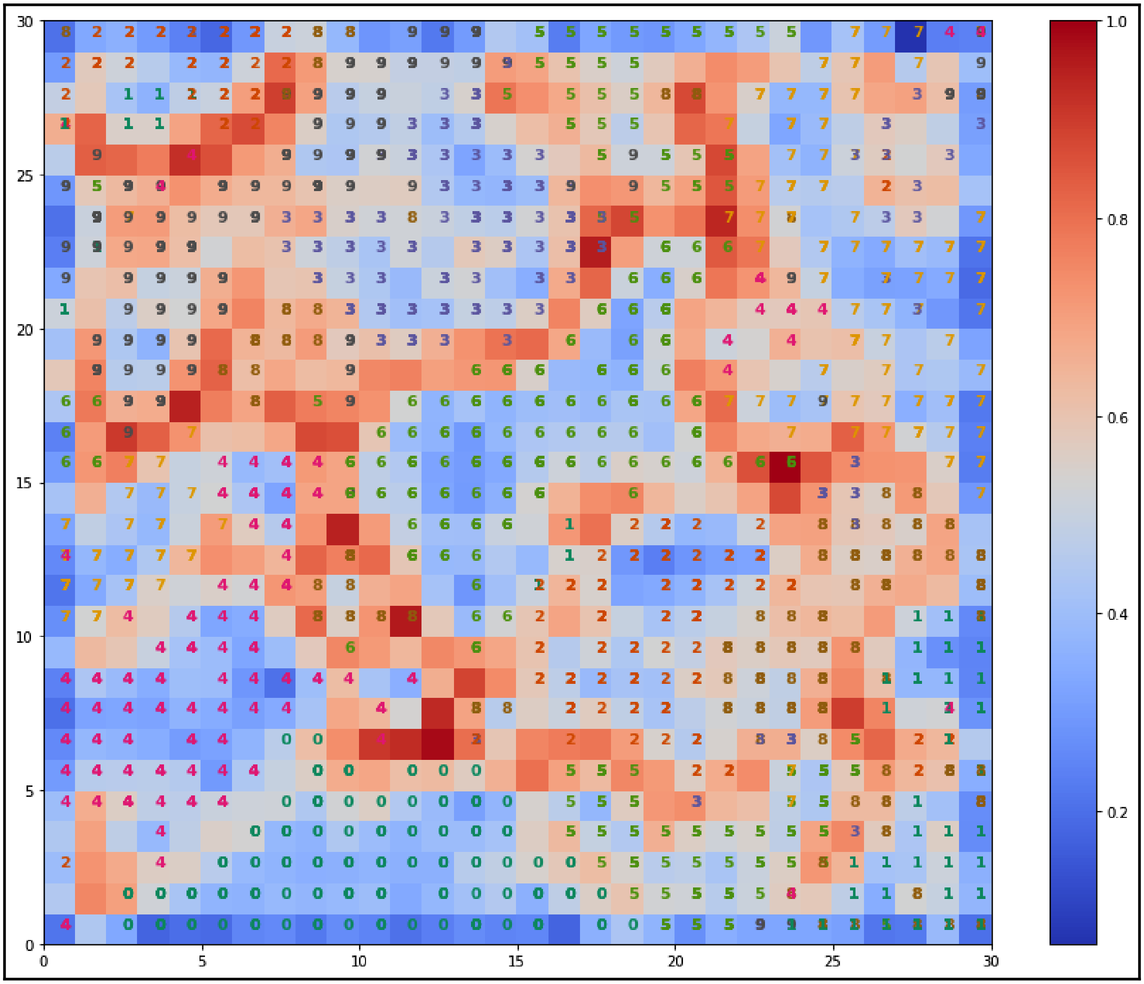
$$\boxed{t_{otsu} = \underset{0 \leq t \leq L-1}{\operatorname{argmin}} \ \sigma_w^2(t)}$$

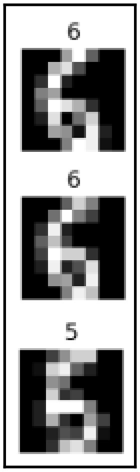

Original
Histogram
otsu
rc
Thresholded (Otsu)
Thresholded (Riddler-Calvard)


Original
Isodata
Li
Mean
Taken from the blackboard of MITx 18.6501x Fundamentals of Statistics class lectures


original
segmented
initial colors
learned colors
segmented (binary)


original
segmented (color-quantized)
initial colors
learned colors





Random Walk for Image Segmentation

Algorithm

1) Map the image intensities to edge weights in the lattice

$$w_{ij} = \exp\left(-\beta(g_i - g_j)^2\right),$$

g_i = image intensity at pixel i

2) Obtain a set, V_M, of marked (labeled) pixels with K labels, either interactively or automatically.

3) Solve for the potentials

$$L_U x^s = -B^T m^s$$

for each label except the final one, f (for computational efficiency). Set $x_i^f = 1 - \sum_{s<f} x_i^s$.

4) Obtain a final segmentation by assigning to each node, v_i, the label corresponding to $\max_s (x_i^s)$.

Combinatorial Dirichlet problem

The **Dirichlet integral** $D[u] = \frac{1}{2}\int_{\Omega} |\nabla u|^2 d\Omega$

harmonic function $\nabla^2 u = 0$

Laplace equation

combinatorial Laplacian matrix

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w_{ij} & \text{if } v_i \text{ and } v_j \text{ are adjacent nodes,} \\ 0 & \text{otherwise,} \end{cases}$$

x_i^s = probability (potential) at node, v_i, for label, s

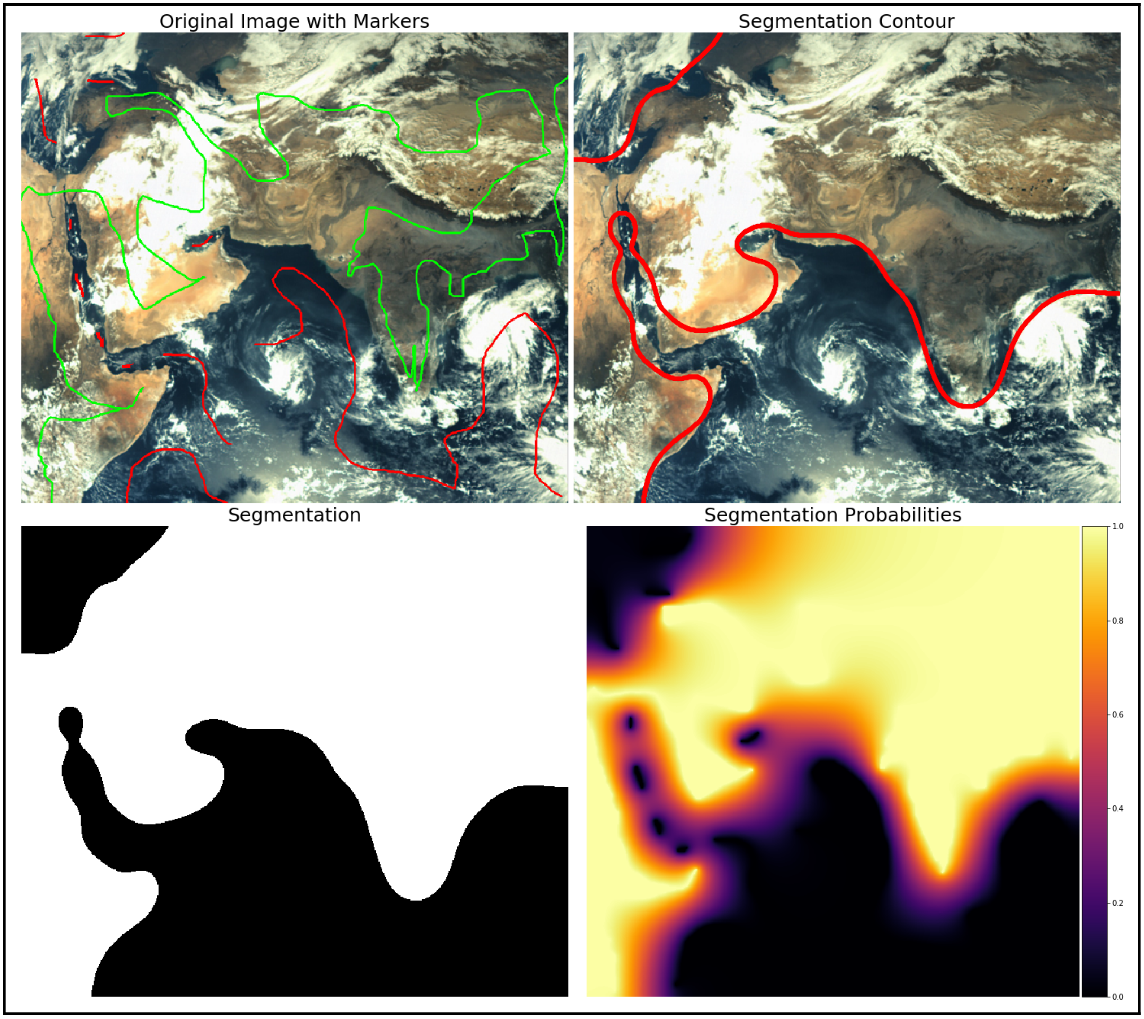
$Q(v_j) = s, \ \forall v_j \in V_M, \ \sum_s x_i^s = 1, \forall v_i \in V$

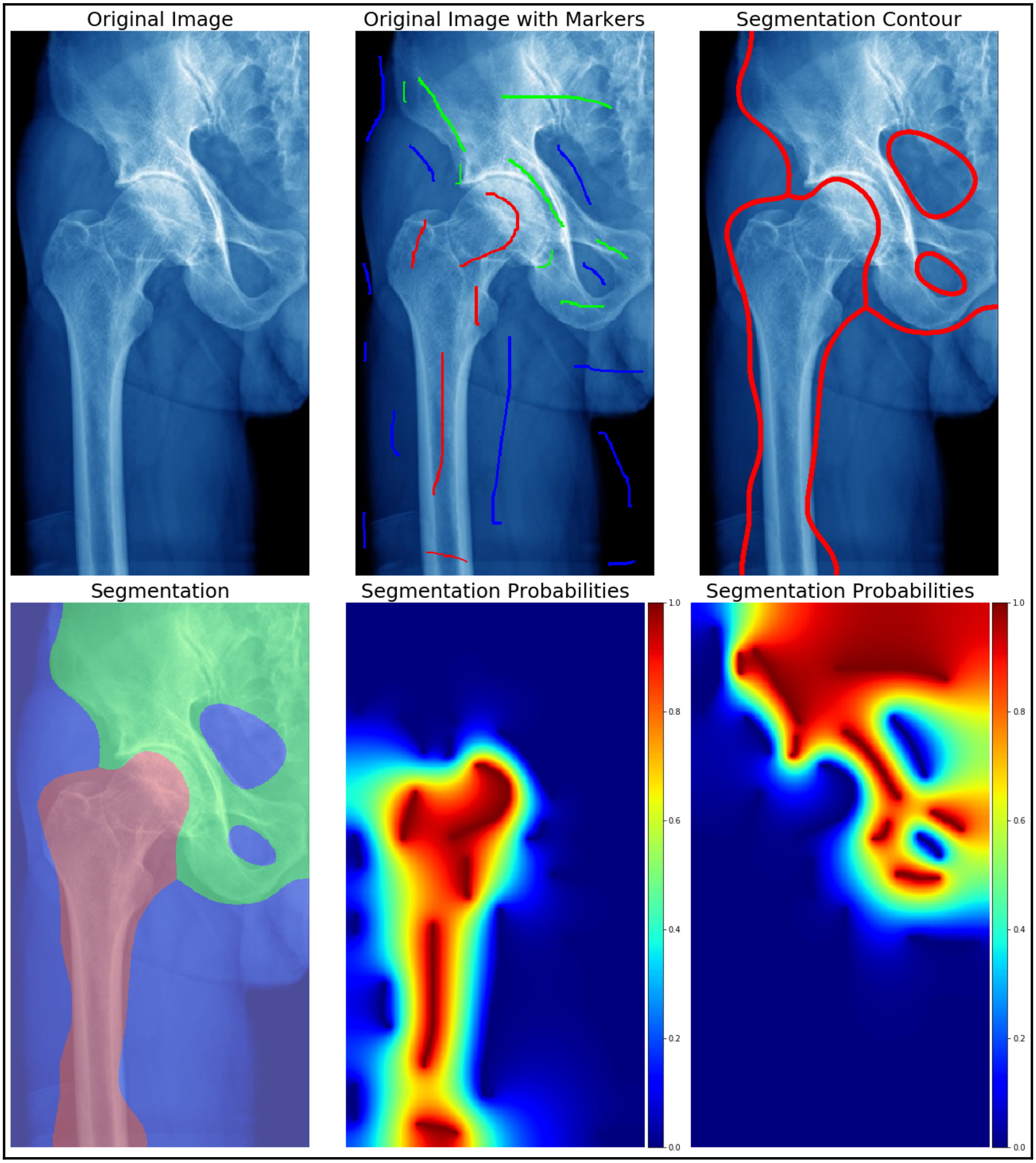
set of labels for seed points

seed points

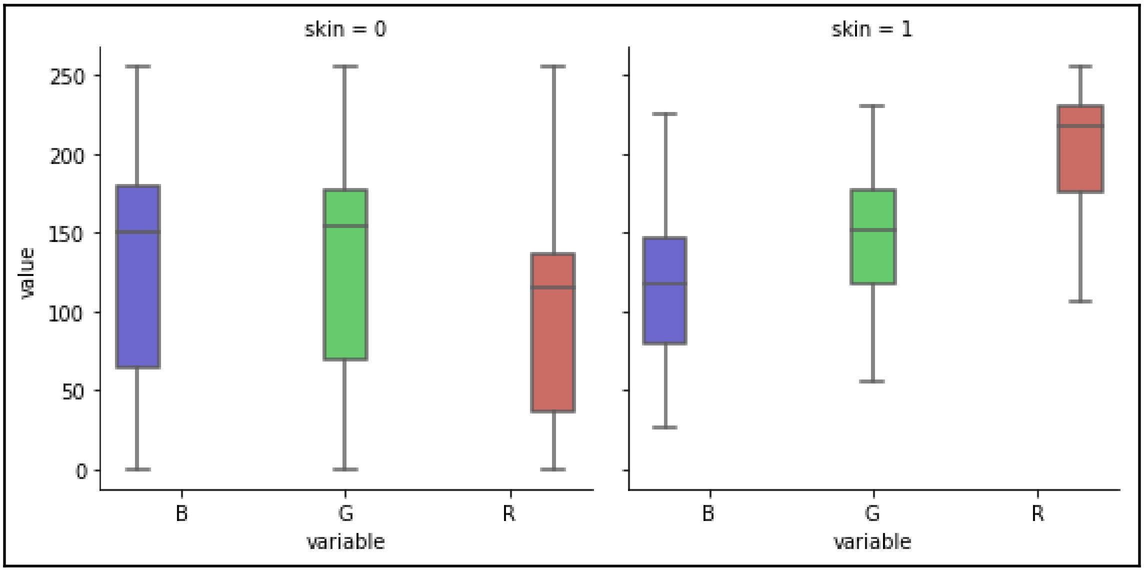
$$m_j^s = \begin{cases} 1 & \text{if } Q(v_j) = s, \\ 0 & \text{if } Q(v_j) \neq s. \end{cases} \qquad s \in \mathbb{Z}, 0 < s \leq K$$

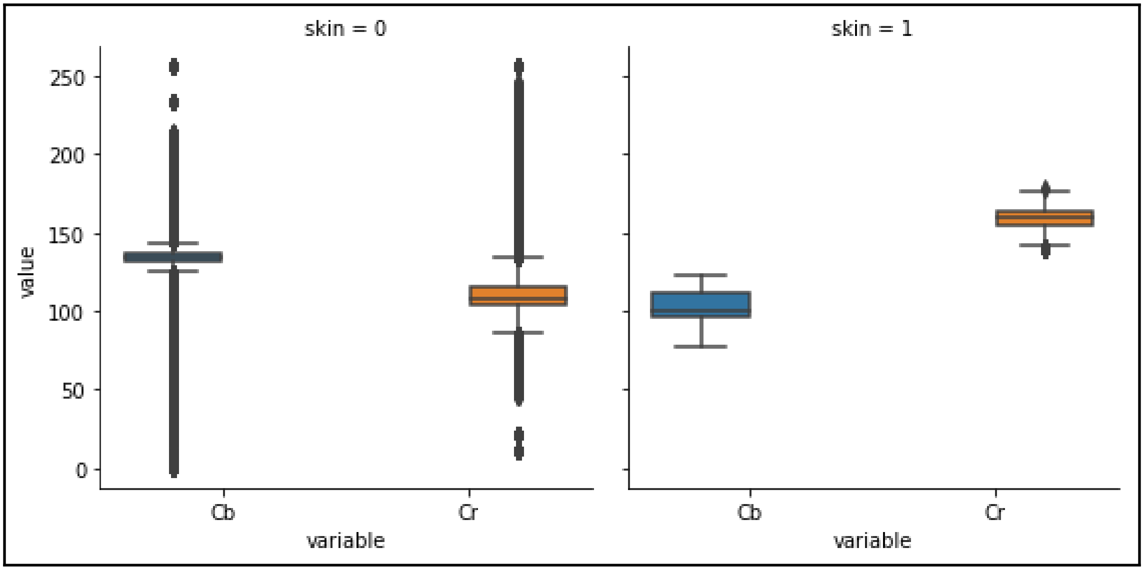
Ref: http://vision.cse.psu.edu/people/chenpingY/paper/grady2006random.pdf

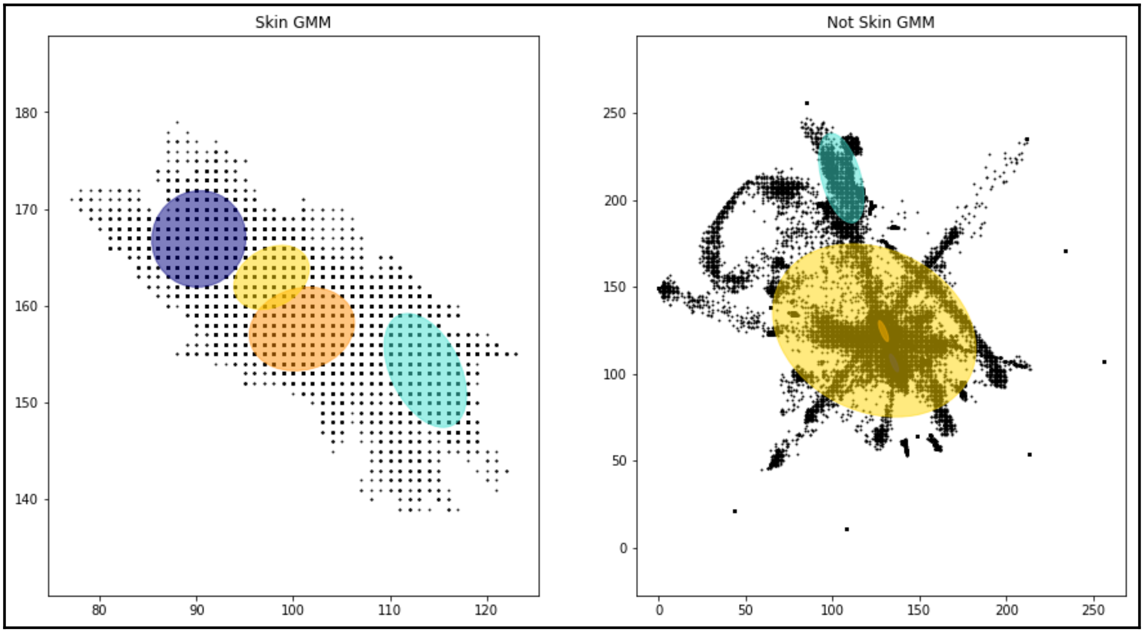

Original Image with Markers
Segmentation Contour
Segmentation
Segmentation Probabilities
1.0
0.8
0.6
0.4
0.2
0.0


Original Image
Original Image with Markers
Segmentation Contour
Segmentation
Segmentation Probabilities
Segmentation Probabilities
1.0
0.8
0.6
0.4
0.2
0.0

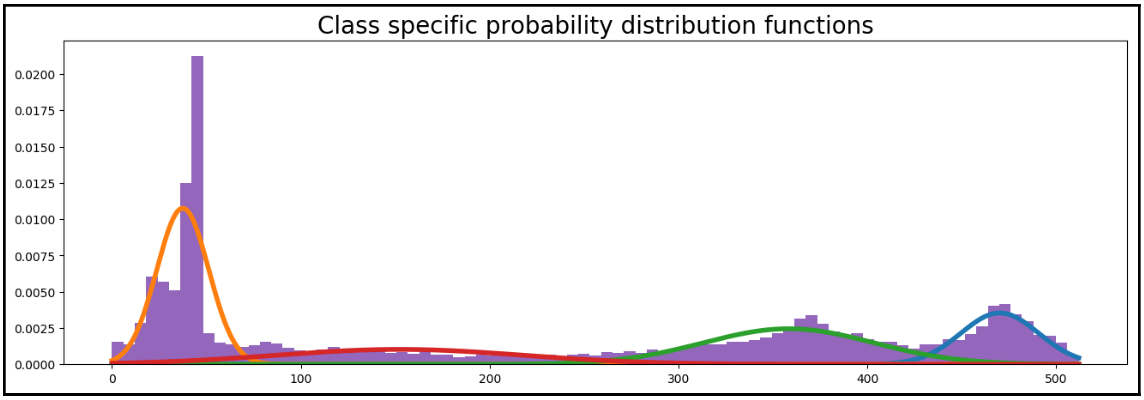
B	G	R	skin
74	85	123	1
73	84	122	1
72	83	121	1
70	81	119	1
70	81	119	1

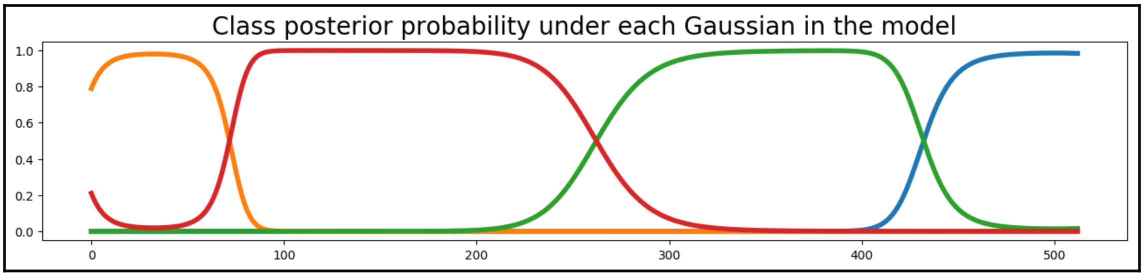


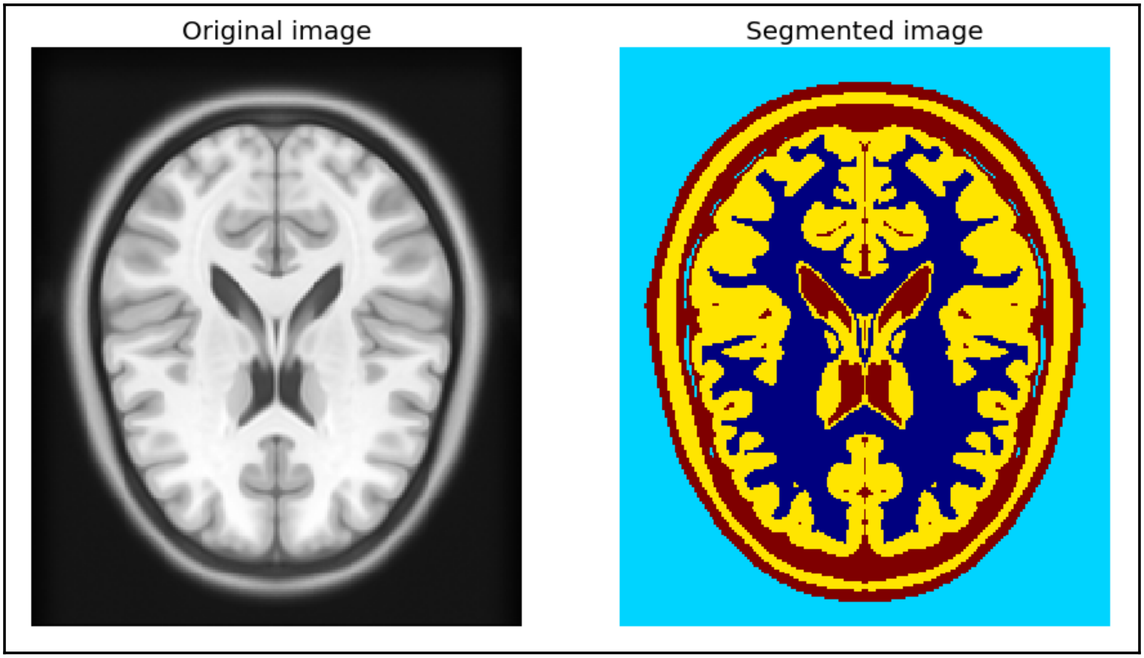

skin = 0
skin = 1
value
250
200
150
100
50
0
Cb
Cr
variable
Cb
Cr
variable

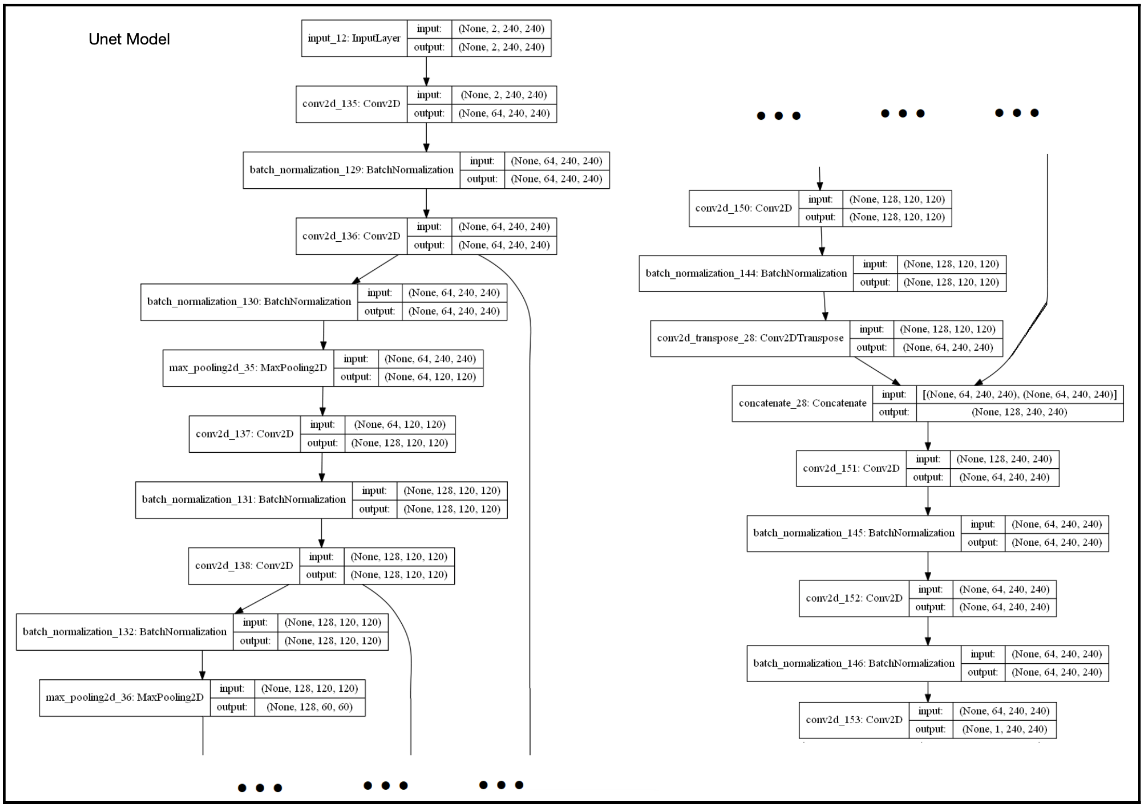

Skin GMM
Not Skin GMM
180
170
160
150
140
80
90
100
110
120
250
200
150
100
50
0
0
50
100
150
200
250

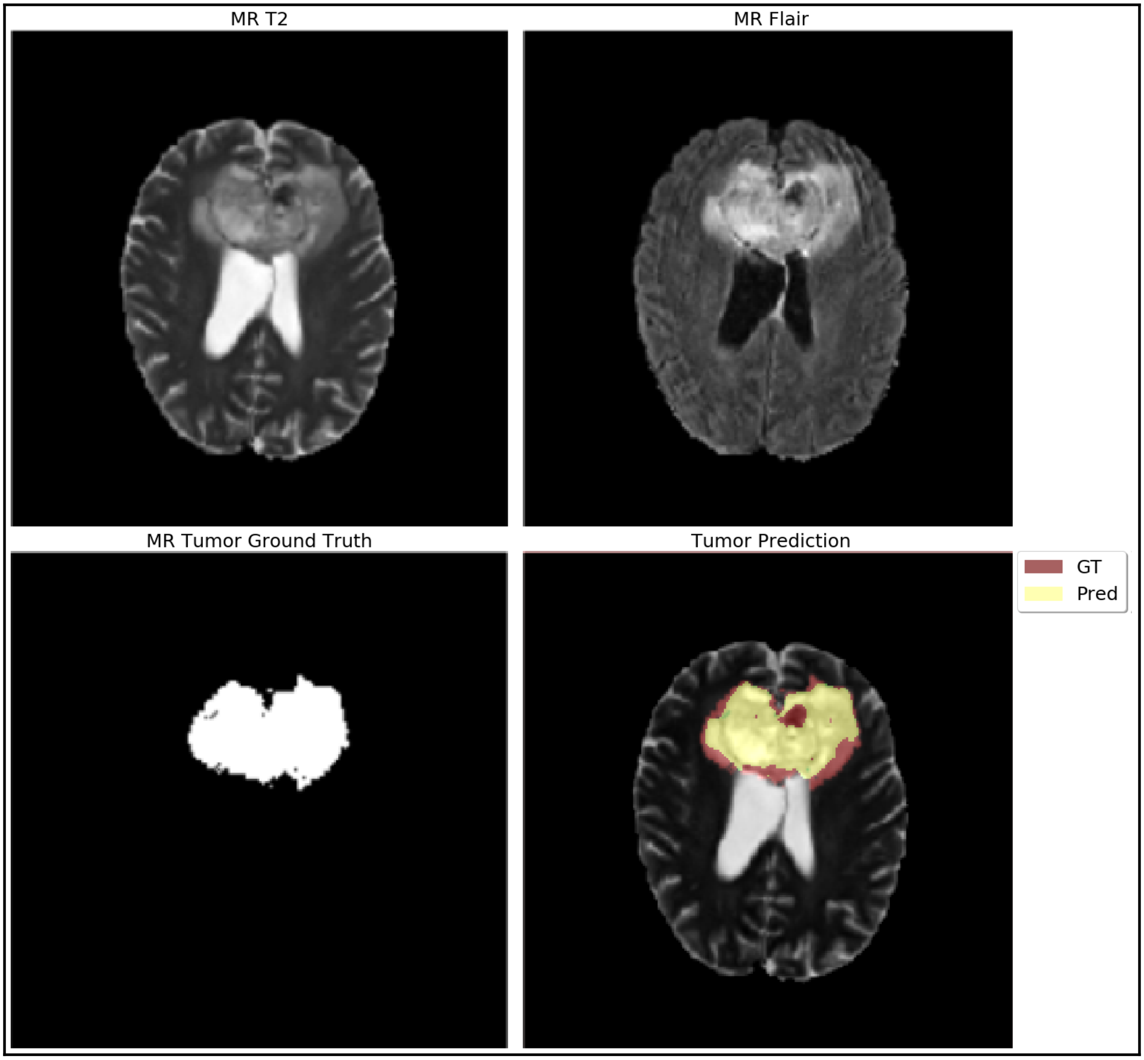

Original
Skin Detected and Segmented with GMM

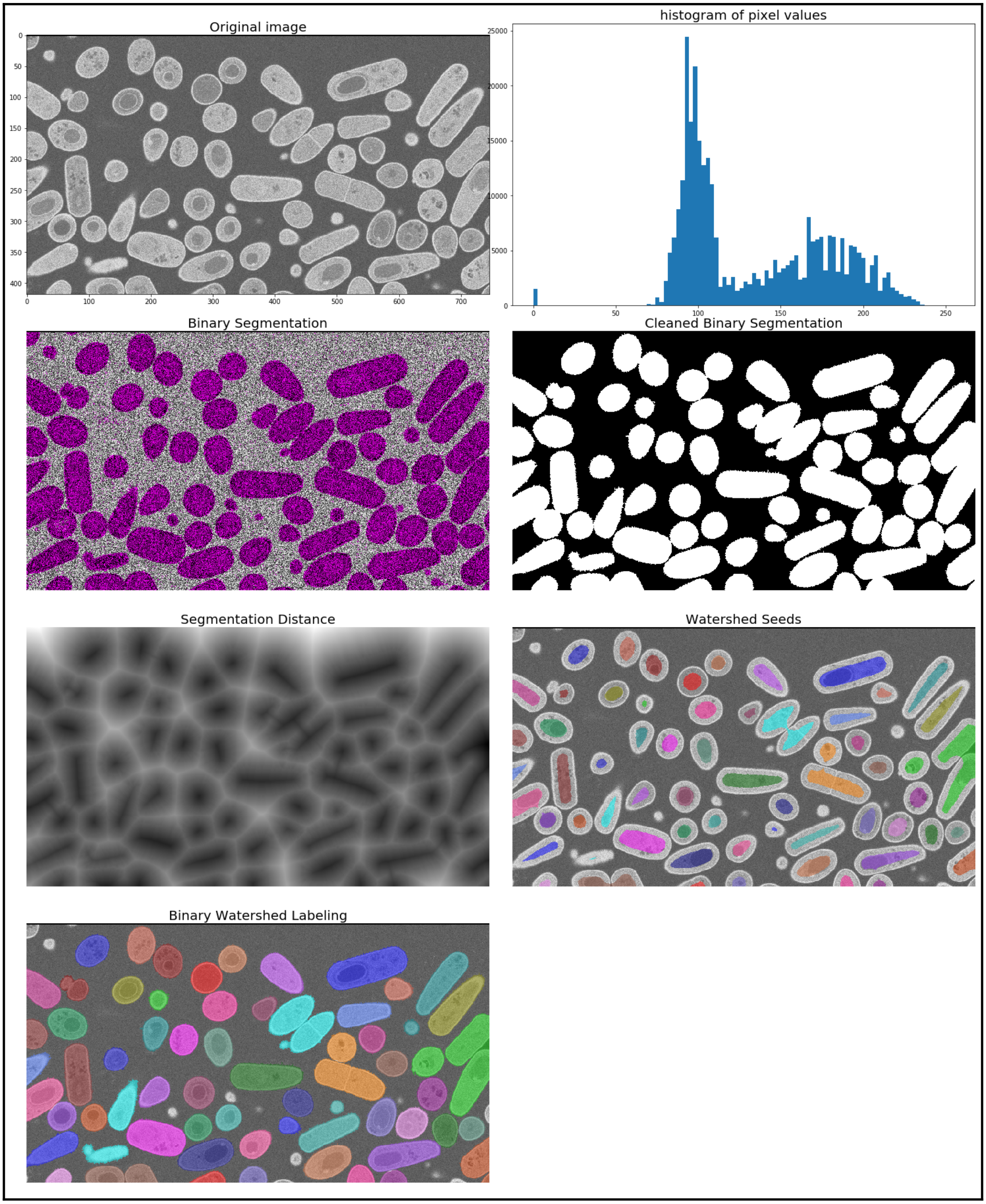

Class specific probability distribution functions
0.0200
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
0
100
200
300
400
500


Class posterior probability under each Gaussian in the model
1.0
0.8
0.6
0.4
0.2
0.0
0
100
200
300
400
500

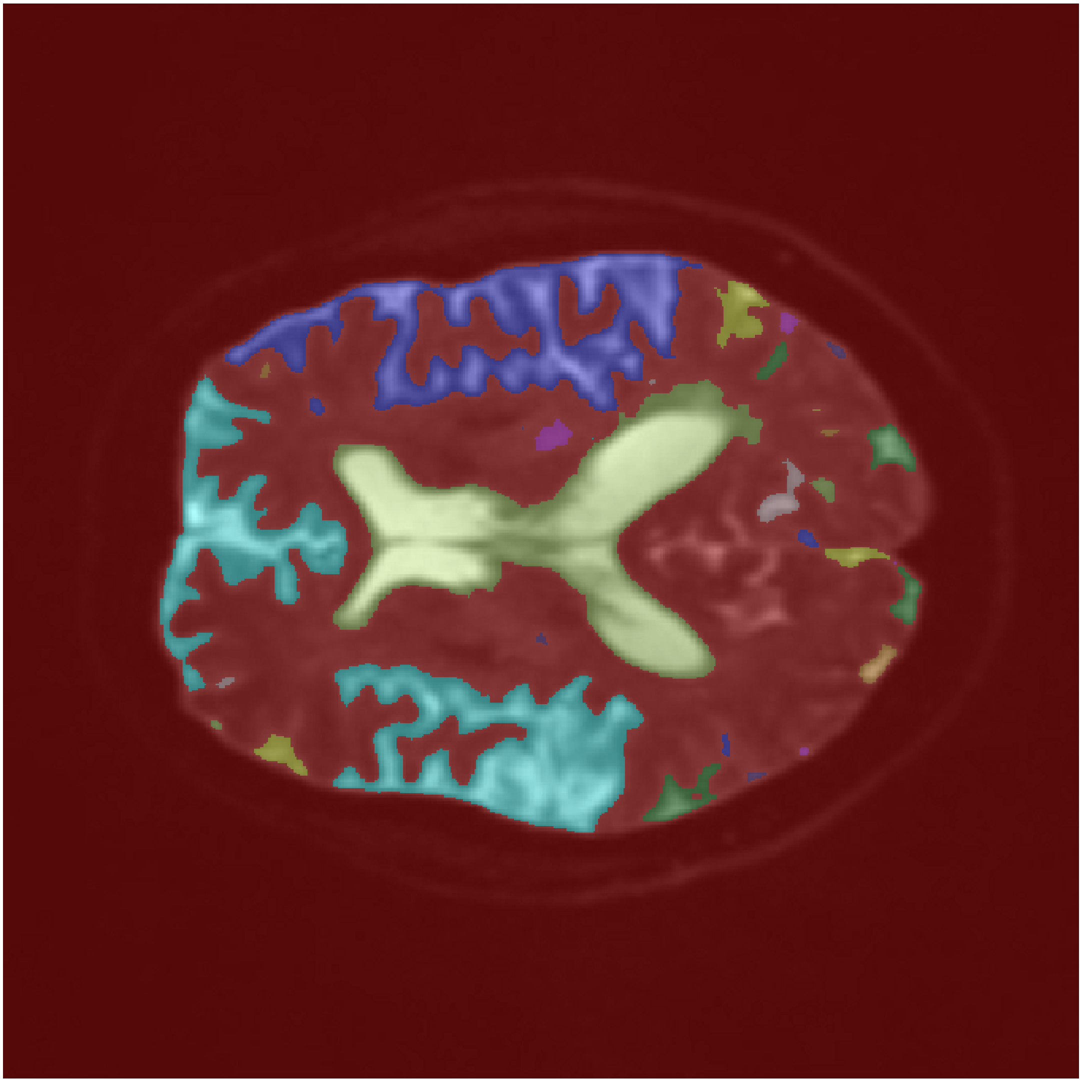

Original image
Segmented image

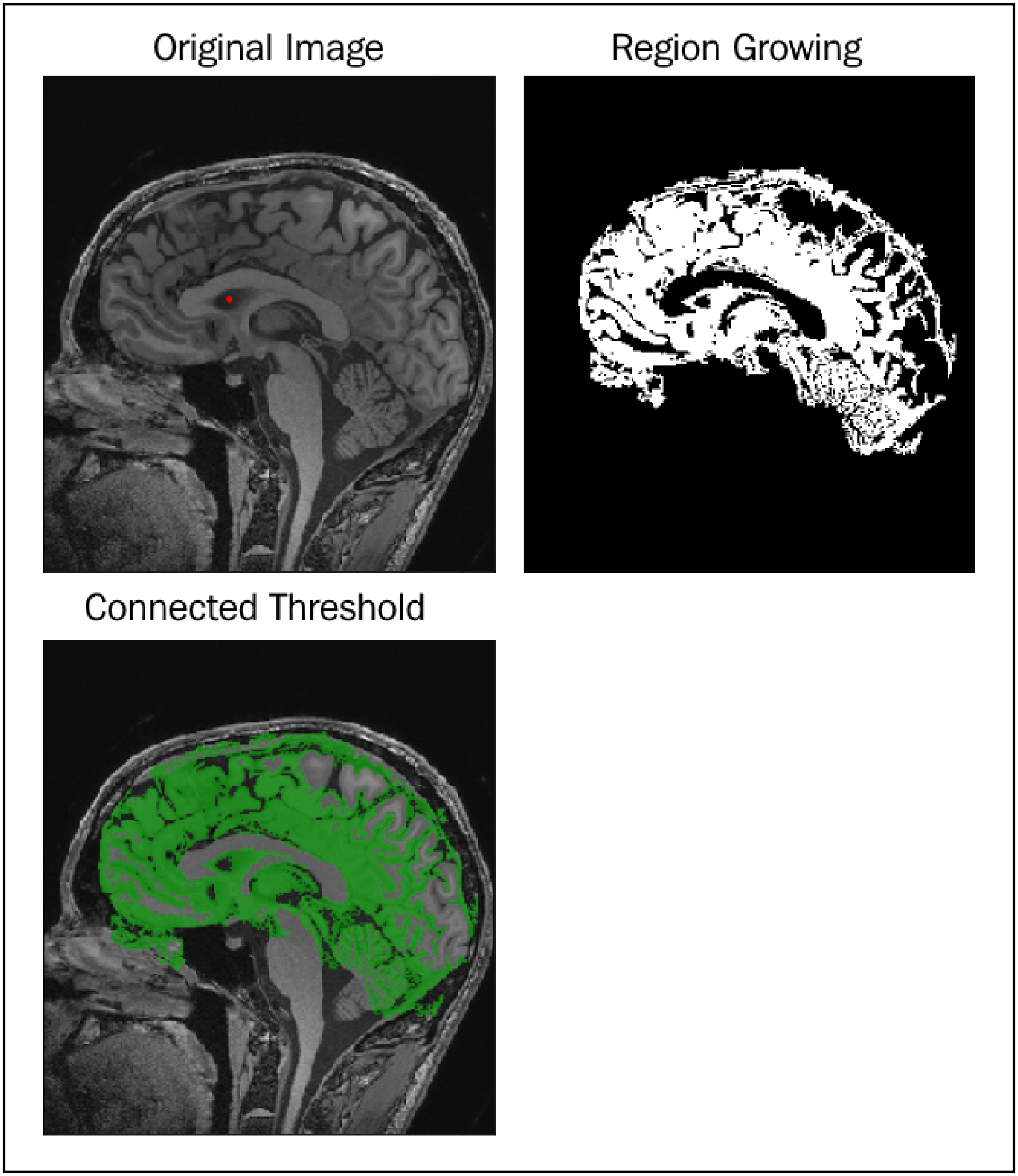

Unet Model
input_12: InputLayer
input: (None, 2, 240, 240)
output: (None, 2, 240, 240)
conv2d_135: Conv2D
input: (None, 2, 240, 240)
output: (None, 64, 240, 240)
batch_normalization_129: BatchNormalization
input: (None, 64, 240, 240)
output: (None, 64, 240, 240)
conv2d_136: Conv2D
input: (None, 64, 240, 240)
output: (None, 64, 240, 240)
batch_normalization_130: BatchNormalization
input: (None, 64, 240, 240)
output: (None, 64, 240, 240)
max_pooling2d_35: MaxPooling2D
input: (None, 64, 240, 240)
output: (None, 64, 120, 120)
conv2d_137: Conv2D
input: (None, 64, 120, 120)
output: (None, 128, 120, 120)
batch_normalization_131: BatchNormalization
input: (None, 128, 120, 120)
output: (None, 128, 120, 120)
conv2d_138: Conv2D
input: (None, 128, 120, 120)
output: (None, 128, 120, 120)
batch_normalization_132: BatchNormalization
input: (None, 128, 120, 120)
output: (None, 128, 120, 120)
max_pooling2d_36: MaxPooling2D
input: (None, 128, 120, 120)
output: (None, 128, 60, 60)
conv2d_150: Conv2D
input: (None, 128, 120, 120)
output: (None, 128, 120, 120)
batch_normalization_144: BatchNormalization
input: (None, 128, 120, 120)
output: (None, 128, 120, 120)
conv2d_transpose_28: Conv2DTranspose
input: (None, 128, 120, 120)
output: (None, 64, 240, 240)
concatenate_28: Concatenate
input: [(None, 64, 240, 240), (None, 64, 240, 240)]
output: (None, 128, 240, 240)
conv2d_151: Conv2D
input: (None, 128, 240, 240)
output: (None, 64, 240, 240)
batch_normalization_145: BatchNormalization
input: (None, 64, 240, 240)
output: (None, 64, 240, 240)
conv2d_152: Conv2D
input: (None, 64, 240, 240)
output: (None, 64, 240, 240)
batch_normalization_146: BatchNormalization
input: (None, 64, 240, 240)
output: (None, 64, 240, 240)
conv2d_153: Conv2D
input: (None, 64, 240, 240)
output: (None, 1, 240, 240)

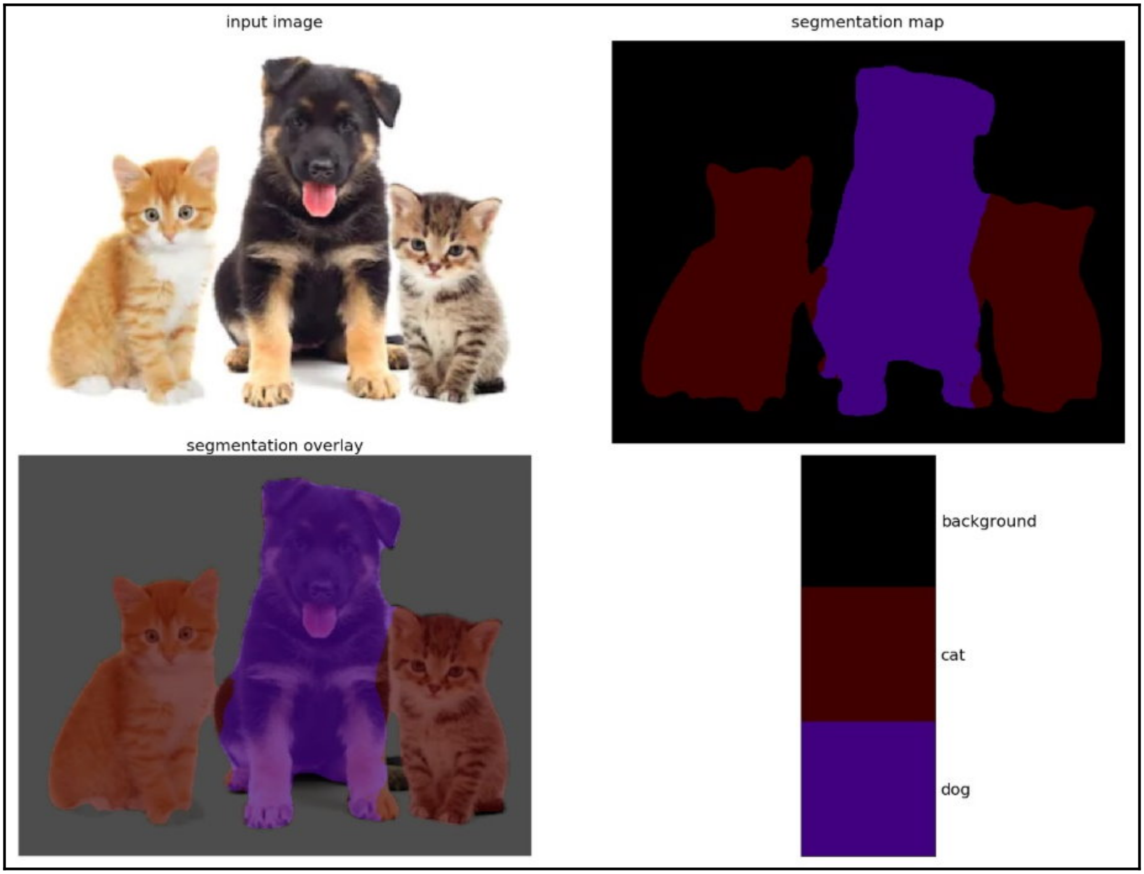

MR T2
MR Flair
MR Tumor Ground Truth
Tumor Prediction
GT
Pred

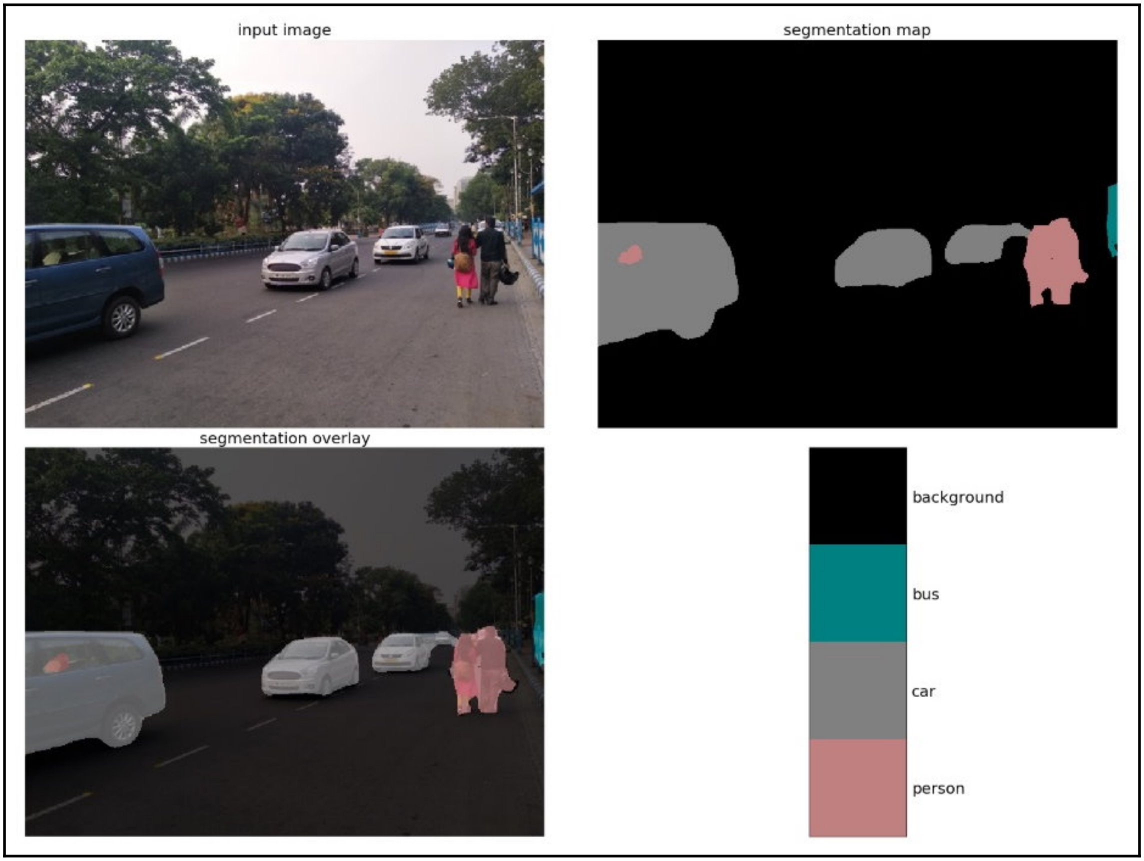


Original image
histogram of pixel values
Binary Segmentation
Cleaned Binary Segmentation
Segmentation Distance
Watershed Seeds
Binary Watershed Labeling

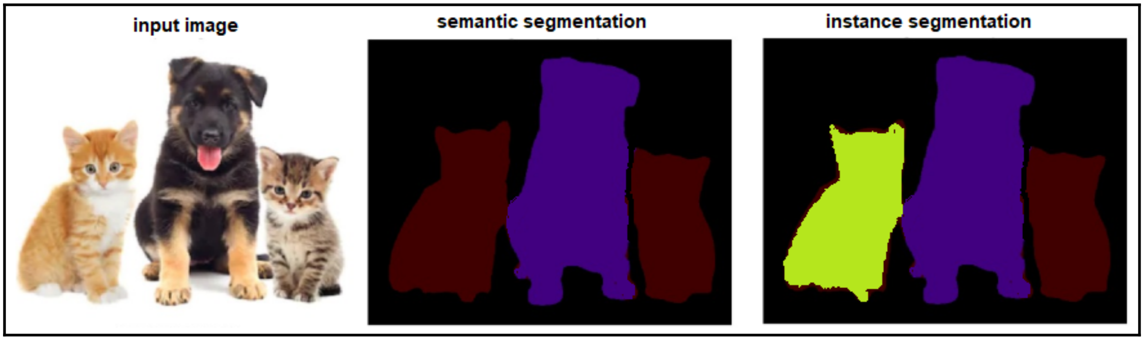


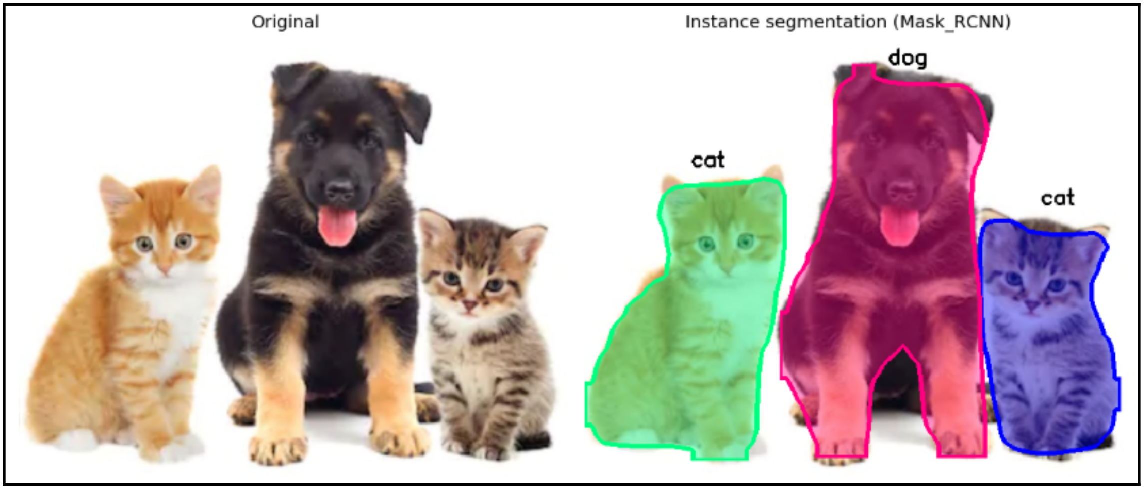

Original Image
Region Growing
Connected Threshold


input image
segmentation map
segmentation overlay
background
cat
dog

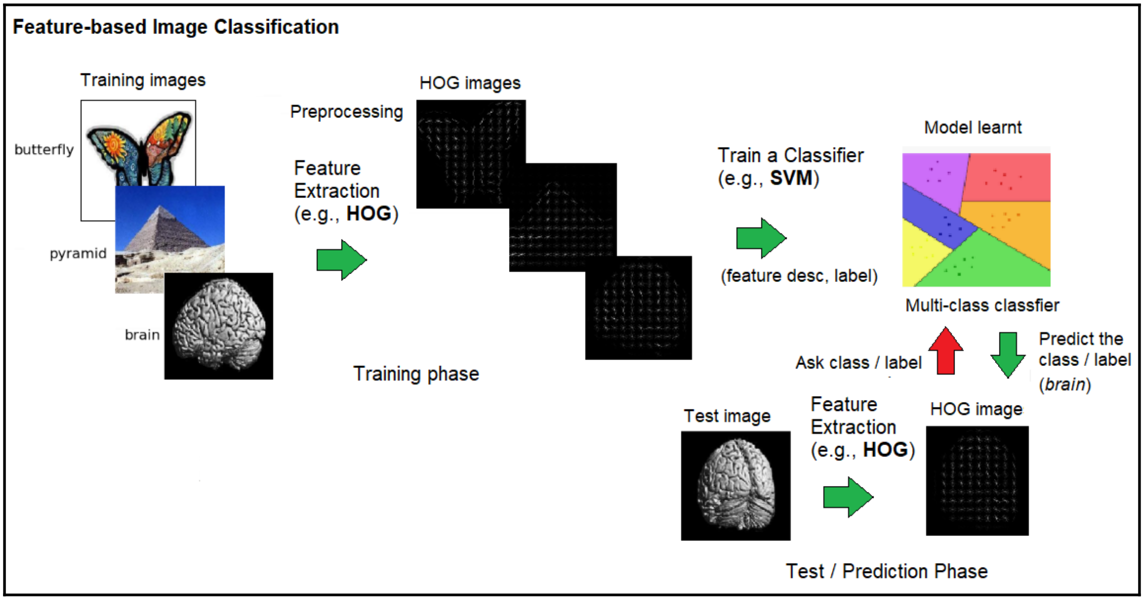

input image
segmentation map
segmentation overlay
background
bus
car
person

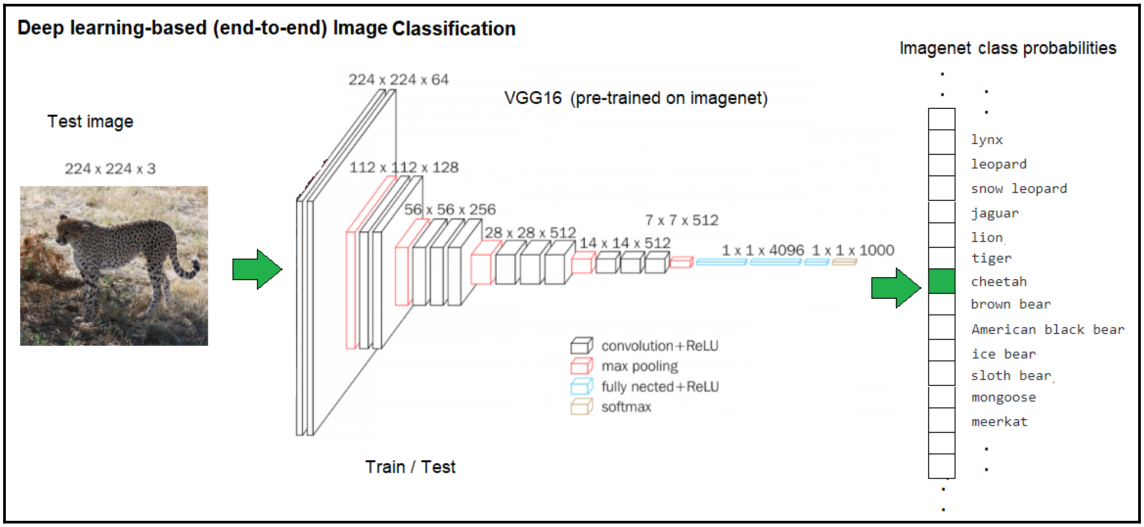

Original Image
Segmentation map
Image with Segmentaion overlay
legends
background
aeroplane
bicycle
bird
boat
bottle
bus
car
cat
chair
cow
diningtable
dog
horse
motorbike
person
pottedplant
sheep
sofa
train
tvmonitor


input image
semantic segmentation
instance segmentation


Original
Instance segmentation (Mask_RCNN)
dog
cat
cat

Chapter 7: Image Classification





Mulitnomial logit Classifier (Softmax Regression)

Training (parameter estimation)

$$J(\theta) = -\left[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\left\{y^{(i)} = k\right\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}\right]$$

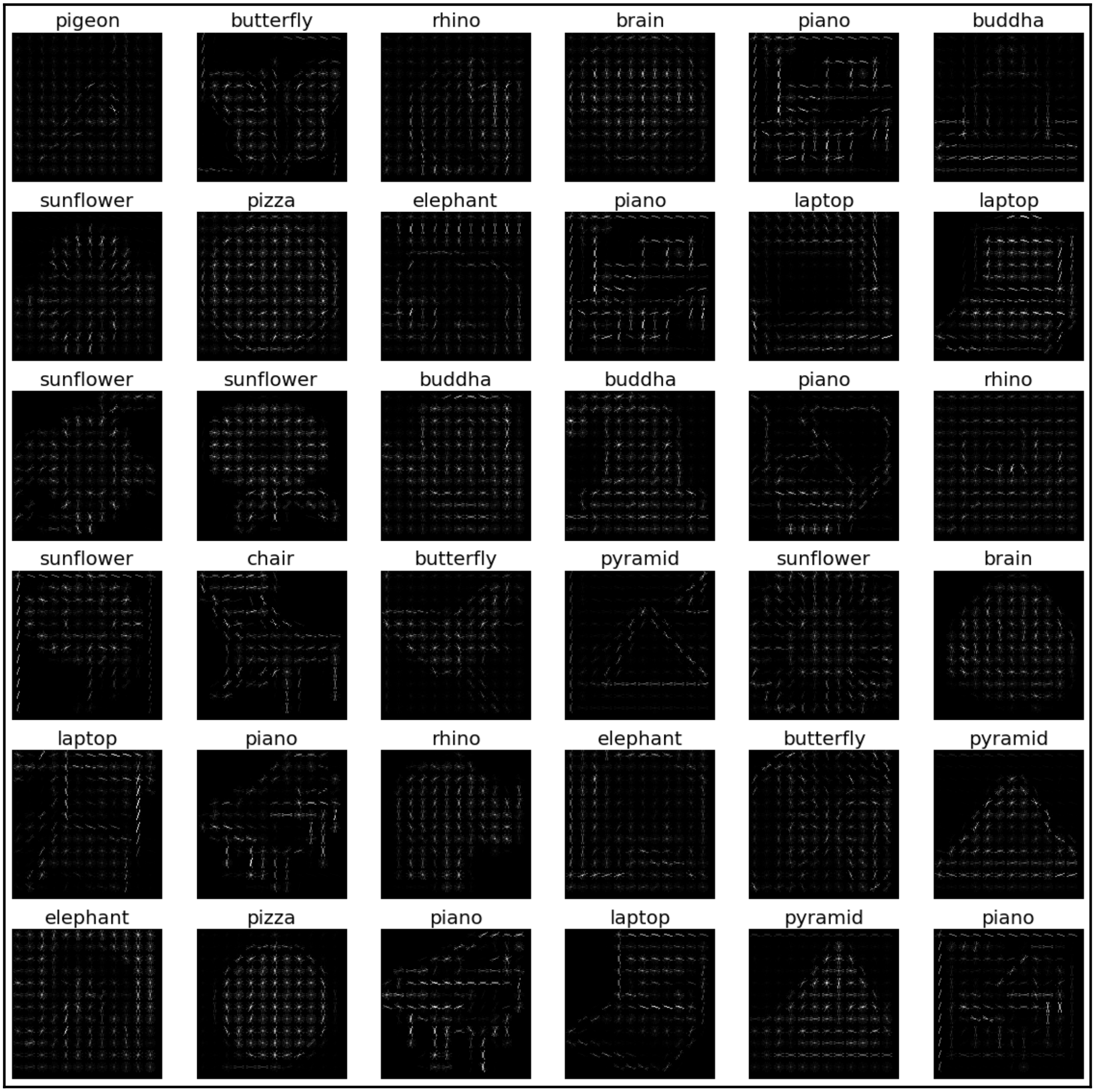
$$\nabla_{\theta^{(k)}} J(\theta) = -\sum_{i=1}^{m}\left[x^{(i)}\left(1\{y^{(i)} = k\} - P(y^{(i)} = k | x^{(i)}; \theta)\right)\right]$$

SGD: $\theta = \theta - \alpha \nabla_{\theta} J(\theta; x^{(i)}, y^{(i)})$

Prediction

$$P(y^{(i)} = k | x^{(i)}; \theta) = \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}$$


pigeon
butterfly
rhino
brain
Neurological Disease Testing Services
piano
buddha
sunflower
pizza
elephant
WALK THIS WAY
piano
laptop
laptop
sunflower
sunflower
buddha
buddha
piano
rhino
sunflower
chair
butterfly
pyramid
sunflower
brain
laptop
piano
rhino
elephant
butterfly
pyramid
elephant
pizza
piano
laptop
pyramid
piano


pigeon
butterfly
rhino
brain
piano
buddha
sunflower
pizza
elephant
piano
laptop
laptop
sunflower
sunflower
buddha
buddha
piano
rhino
sunflower
chair
butterfly
pyramid
sunflower
brain
laptop
piano
rhino
elephant
butterfly
pyramid
elephant
pizza
piano
laptop
pyramid
piano

```
Accuracy: 0.8068181818181818
              precision    recall  f1-score   support

       brain       0.90      0.75      0.82        12
      buddha       0.75      0.60      0.67         5
   butterfly       0.88      0.78      0.82         9
       chair       1.00      0.25      0.40         4
    elephant       0.78      0.88      0.82         8
      laptop       0.88      1.00      0.93         7
       piano       1.00      1.00      1.00        12
      pigeon       1.00      0.60      0.75         5
       pizza       0.38      0.75      0.50         4
     pyramid       0.60      0.75      0.67         4
       rhino       0.70      1.00      0.82         7
   sunflower       0.90      0.82      0.86        11

    accuracy                           0.81        88
   macro avg       0.81      0.76      0.76        88
weighted avg       0.85      0.81      0.81        88
```


Actual vs. Predicted Class Labels
brain/brain
piano/piano
butterfly/butterfly
sunflower/sunflower
sunflower/sunflower
chair/chair
sunflower/sunflower
brain/pizza
butterfly/butterfly
piano/piano
laptop/laptop
rhino/rhino
sunflower/elephant
brain/sunflower
sunflower/sunflower
rhino/rhino
pigeon/rhino
elephant/elephant
butterfly/pizza
pyramid/brain
piano/piano
chair/rhino
sunflower/sunflower
pyramid/pyramid
sunflower/sunflower
elephant/elephant
pizza/pizza
sunflower/sunflower
brain/pizza
elephant/elephant
laptop/laptop
brain/brain
piano/piano
buddha/buddha
butterfly/pizza
laptop/laptop
pigeon/pigeon
laptop/laptop
piano/piano
buddha/buddha
rhino/rhino
pizza/pizza
butterfly/butterfly
rhino/rhino
piano/piano
buddha/buddha
laptop/laptop
elephant/elephant
elephant/rhino
elephant/elephant
pyramid/pyramid
piano/piano
brain/brain
brain/brain
pizza/butterfly
rhino/rhino
brain/brain
brain/brain
piano/piano
butterfly/butterfly
pigeon/pigeon
chair/pizza
pigeon/buddha
pyramid/pyramid
brain/brain
piano/piano
butterfly/butterfly
rhino/rhino
piano/piano
pizza/pizza
elephant/elephant
butterfly/butterfly
sunflower/sunflower
buddha/pyramid
sunflower/elephant
piano/piano
rhino/rhino
pigeon/pigeon
laptop/laptop
sunflower/sunflower
chair/laptop
buddha/pyramid
brain/brain
piano/piano
butterfly/butterfly
brain/brain
elephant/elephant
laptop/laptop

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x'}{\lambda} + \psi\right)\right)$$

where $x' = x\cos\theta + y\sin\theta$

$y' = -x\sin\theta + y\cos\theta$

λ wavelength of the sinusoidal factor

θ orientation of the normal to the parallel stripes of a Gabor function

ψ phase offset

σ standard deviation of the Gaussian envelope

γ spatial aspect ratio

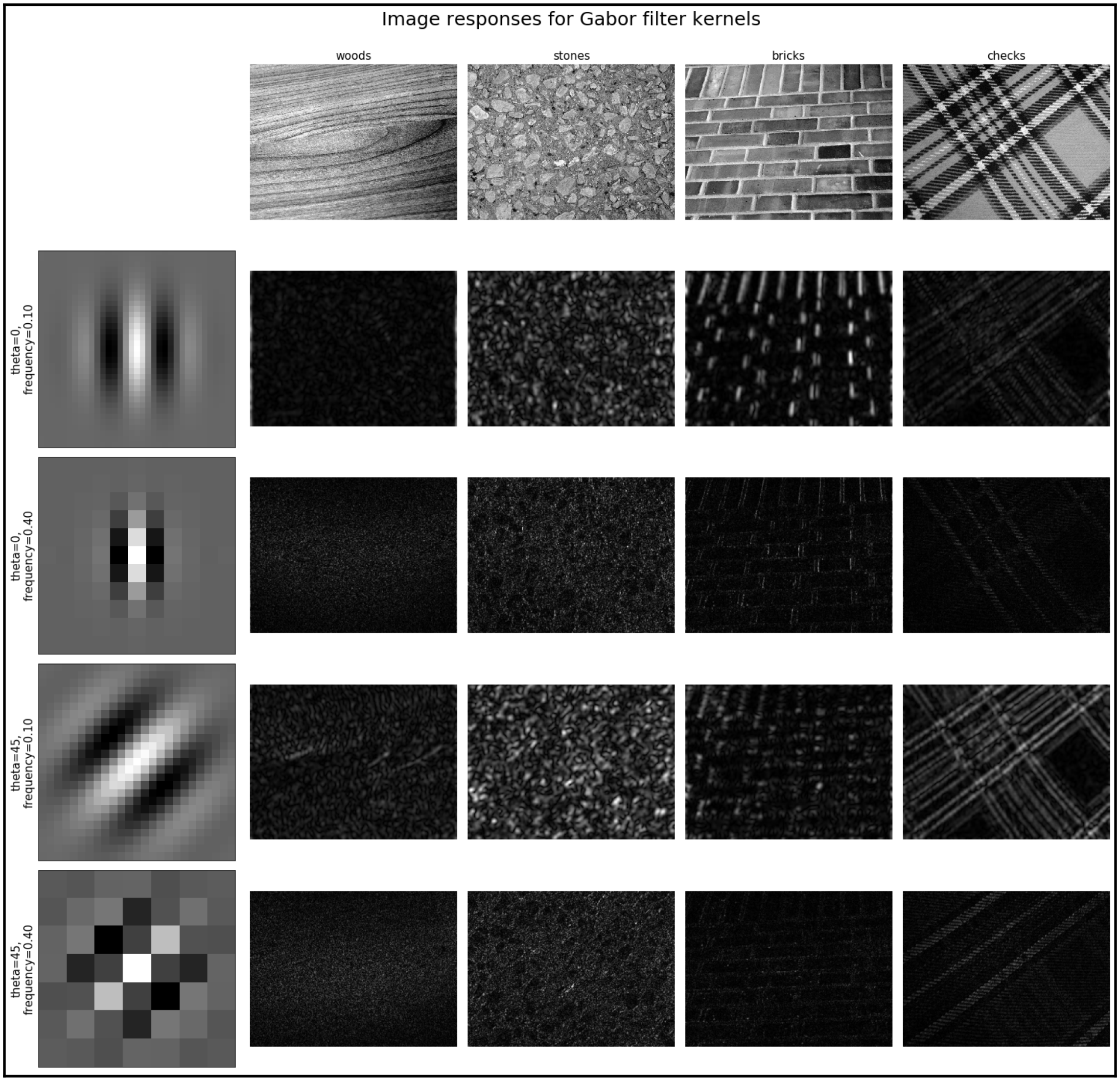
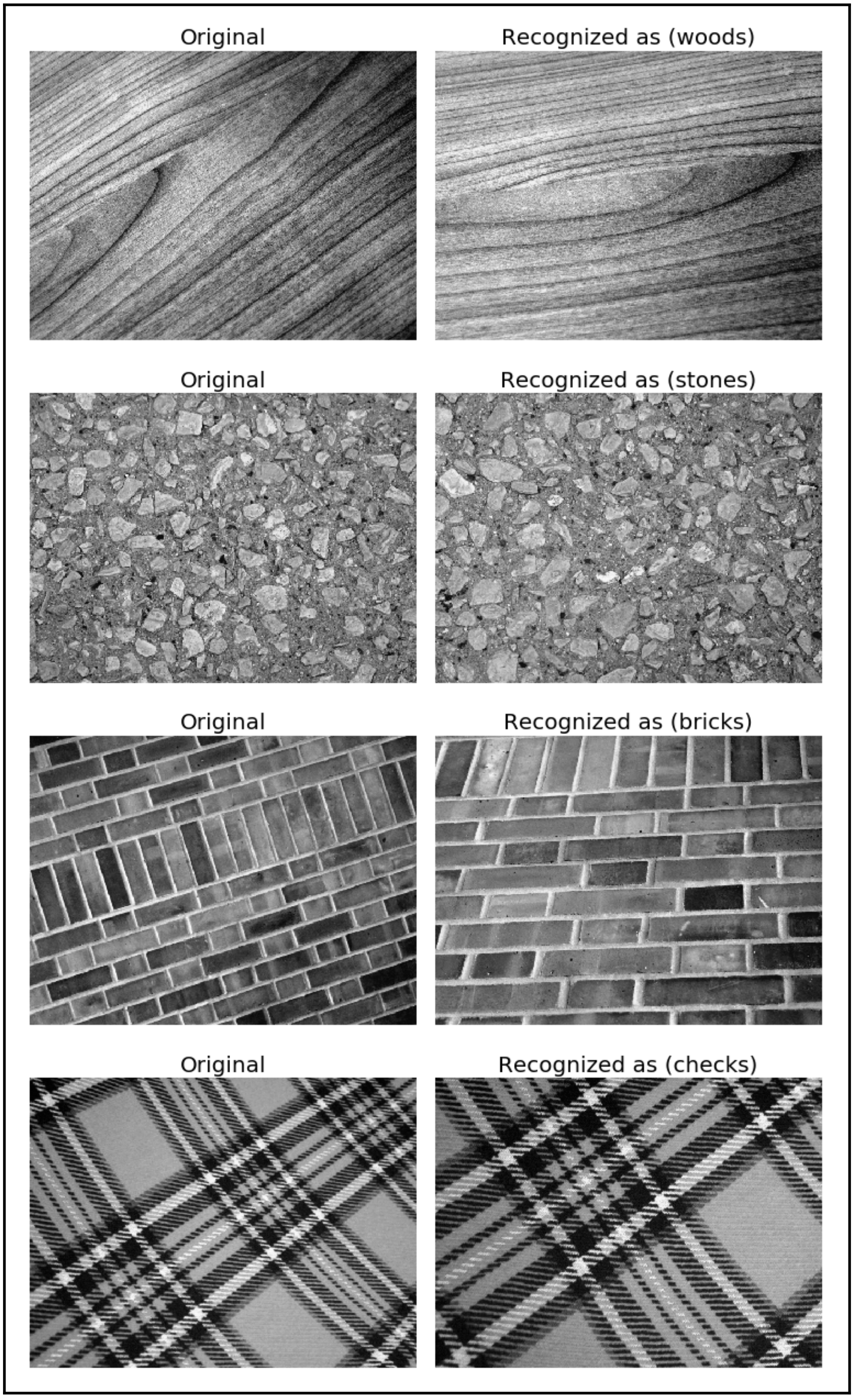
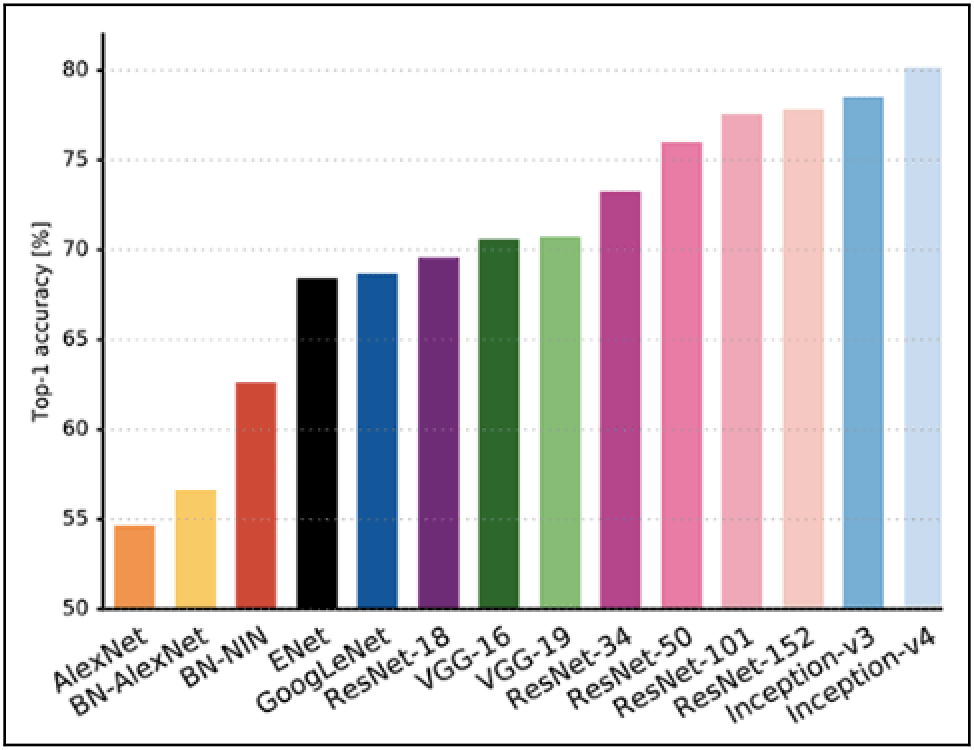


Image responses for Gabor filter kernels
woods
stones
bricks
checks
theta=0,
frequency=0.10
theta=0,
frequency=0.40
theta=45,
frequency=0.10
theta=45,
frequency=0.40


Original
Recognized as (woods)
Original
Recognized as (stones)
Original
Recognized as (bricks)
Original
Recognized as (checks)



Top-1 accuracy [%]
80
75
70
65
60
55
50
AlexNet
BN-AlexNet
BN-NIN
ENet
GoogLeNet
ResNet-18
VGG-16
VGG-19
ResNet-34
ResNet-50
ResNet-101
ResNet-152
Inception-v3
Inception-v4

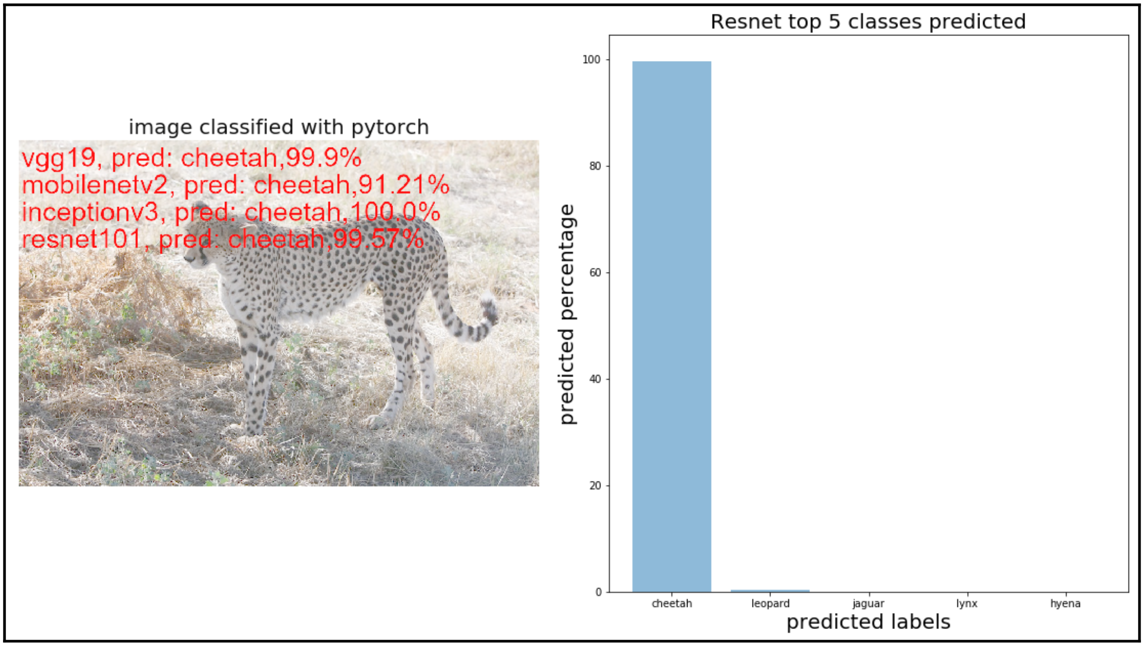


image classified with pytorch
vgg19, pred: cheetah,99.9%
mobilenetv2, pred: cheetah,91.21%
inceptionv3, pred: cheetah,100.0%
resnet101, pred: cheetah,99.57%
Resnet top 5 classes predicted
predicted percentage
0
20
40
60
80
100
cheetah
leopard
jaguar
lynx
hyena
predicted labels

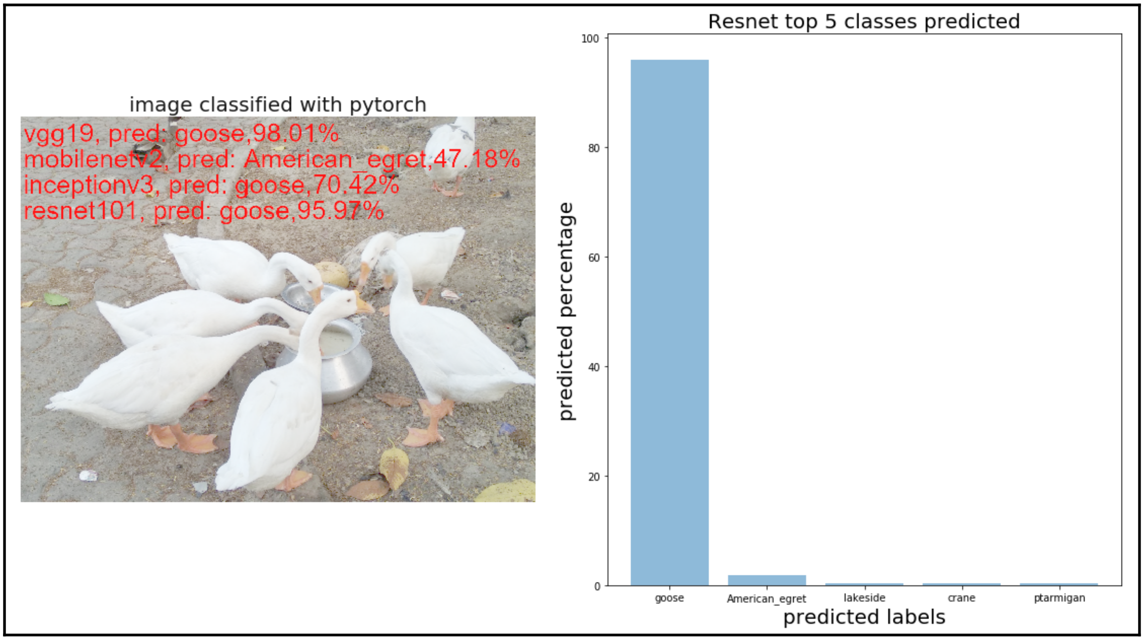
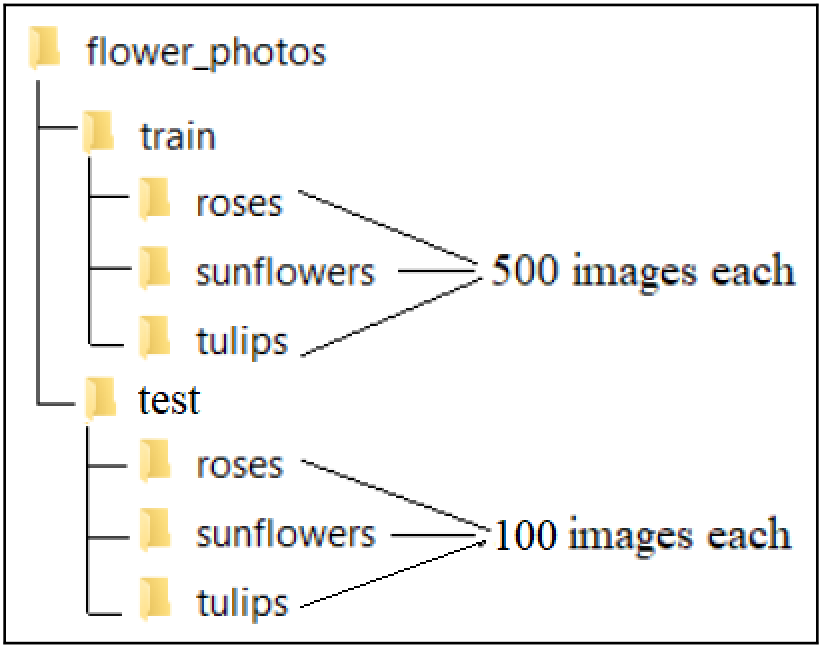
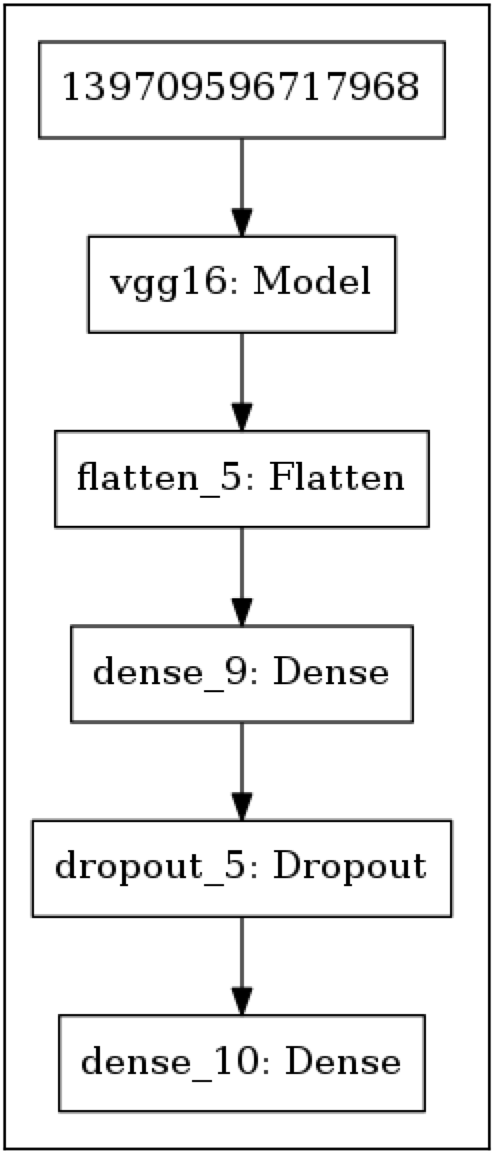
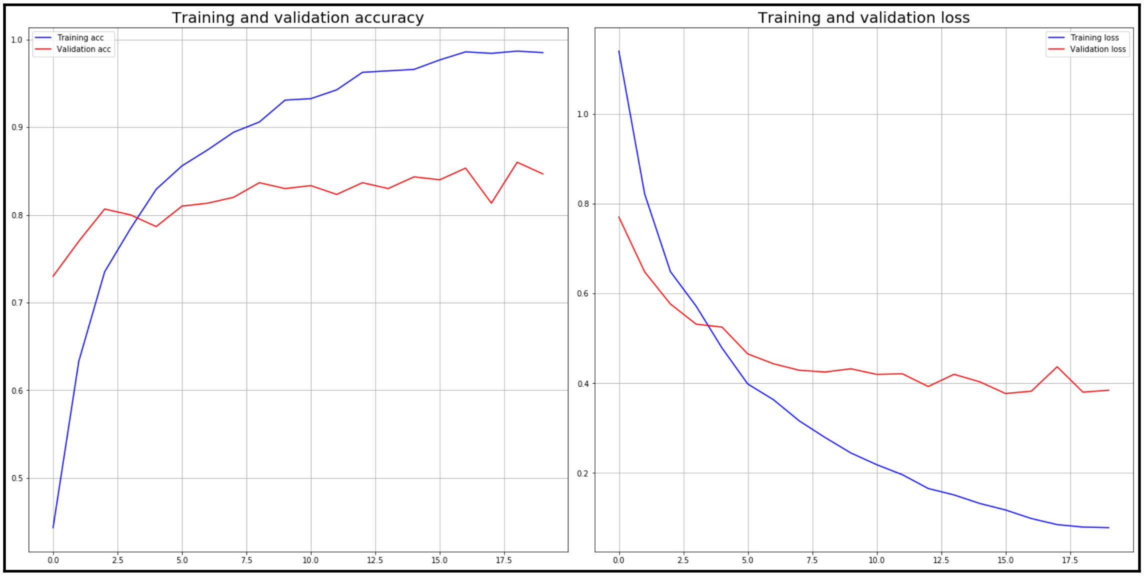


image classified with pytorch
vgg19, pred: goose,98.01%
mobilenetv2, pred: American_egret,47.18%
inceptionv3, pred: goose,70.42%
resnet101, pred: goose,95.97%
Resnet top 5 classes predicted
predicted percentage
0
20
40
60
80
100
goose
American_egret
lakeside
crane
ptarmigan
predicted labels

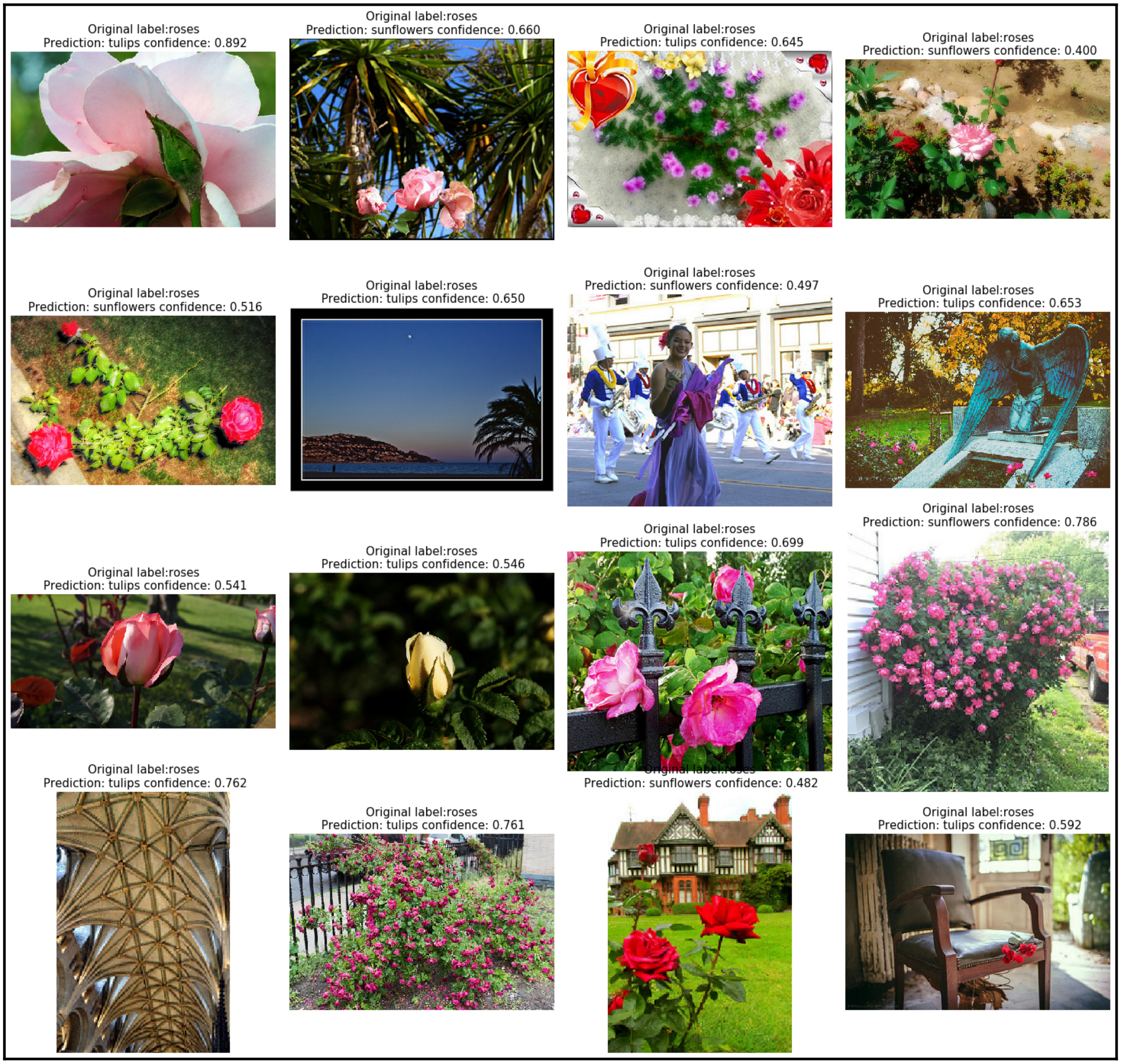

VGG16: flamingo, 0.83
VGG16: daisy, 0.43
VGG16: artichoke, 0.33


flower_photos
train
roses
sunflowers
tulips
500 images each
test
roses
sunflowers
tulips
100 images each

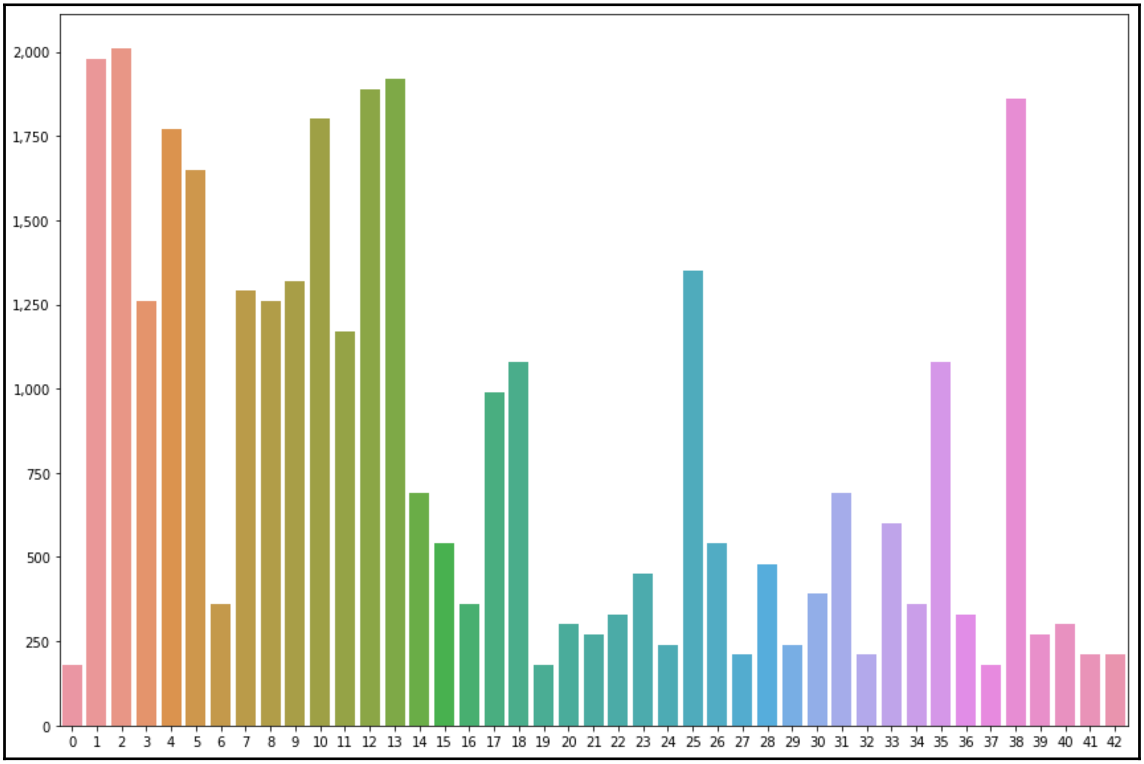

139709596717968
vgg16: Model
flatten_5: Flatten
dense_9: Dense
dropout_5: Dropout
dense_10: Dense

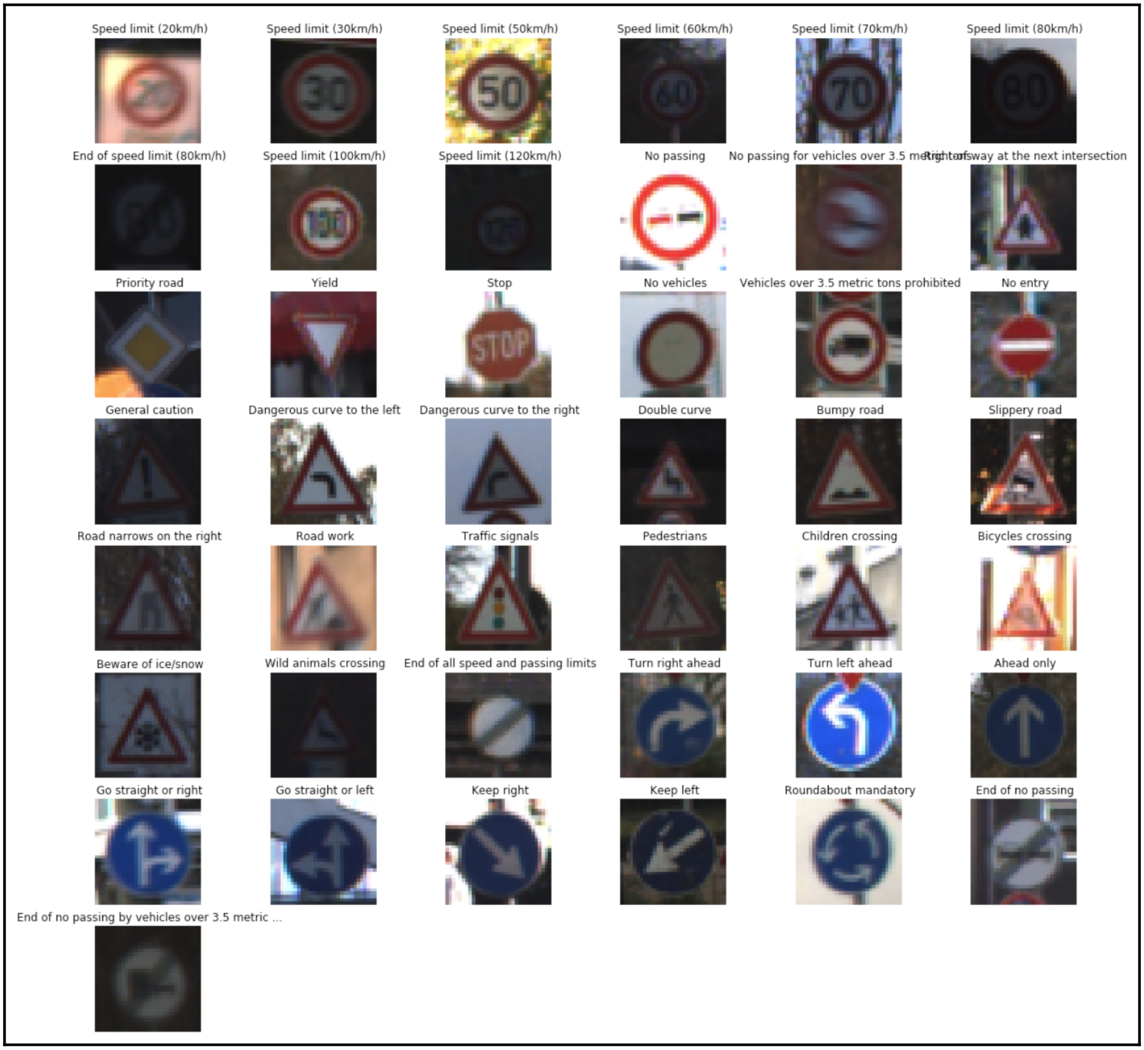


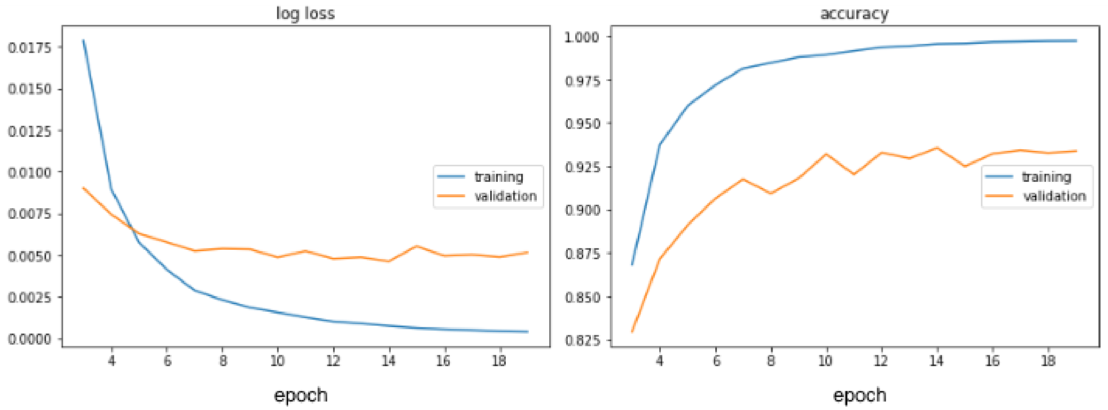
Training and validation accuracy
Training acc
Validation acc
Training and validation loss
Training loss
Validation loss



	ClassId	SignName
0	0	Speed limit (20km/h)
1	1	Speed limit (30km/h)
2	2	Speed limit (50km/h)
3	3	Speed limit (60km/h)
4	4	Speed limit (70km/h)

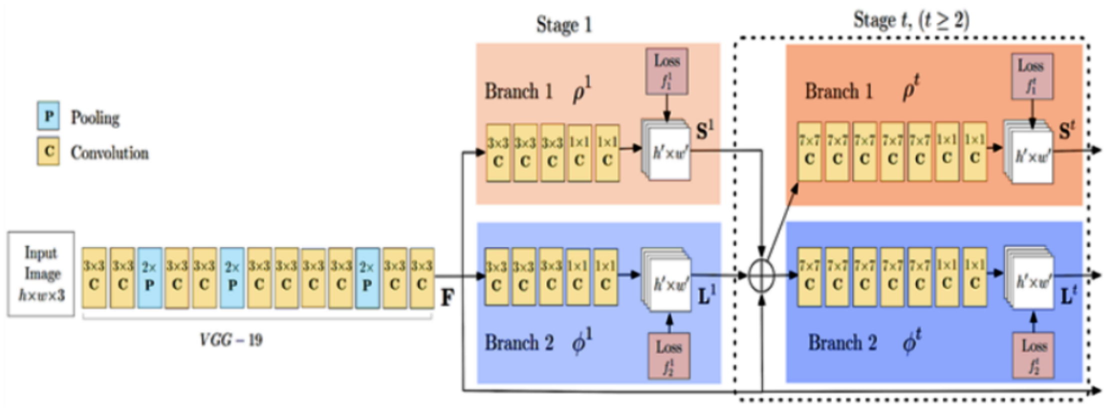

2,000
1,750
1,500
1,250
1,000
750
500
250
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42


Speed limit (20km/h)
Speed limit (30km/h)
Speed limit (50km/h)
Speed limit (60km/h)
Speed limit (70km/h)
Speed limit (80km/h)
End of speed limit (80km/h)
Speed limit (100km/h)
Speed limit (120km/h)
No passing
No passing for vehicles over 3.5 metric tons
Right-of-way at the next intersection
Priority road
Yield
Stop
No vehicles
Vehicles over 3.5 metric tons prohibited
No entry
General caution
Dangerous curve to the left
Dangerous curve to the right
Double curve
Bumpy road
Slippery road
Road narrows on the right
Road work
Traffic signals
Pedestrians
Children crossing
Bicycles crossing
Beware of ice/snow
Wild animals crossing
End of all speed and passing limits
Turn right ahead
Turn left ahead
Ahead only
Go straight or right
Go straight or left
Keep right
Keep left
Roundabout mandatory
End of no passing
End of no passing by vehicles over 3.5 metric ...



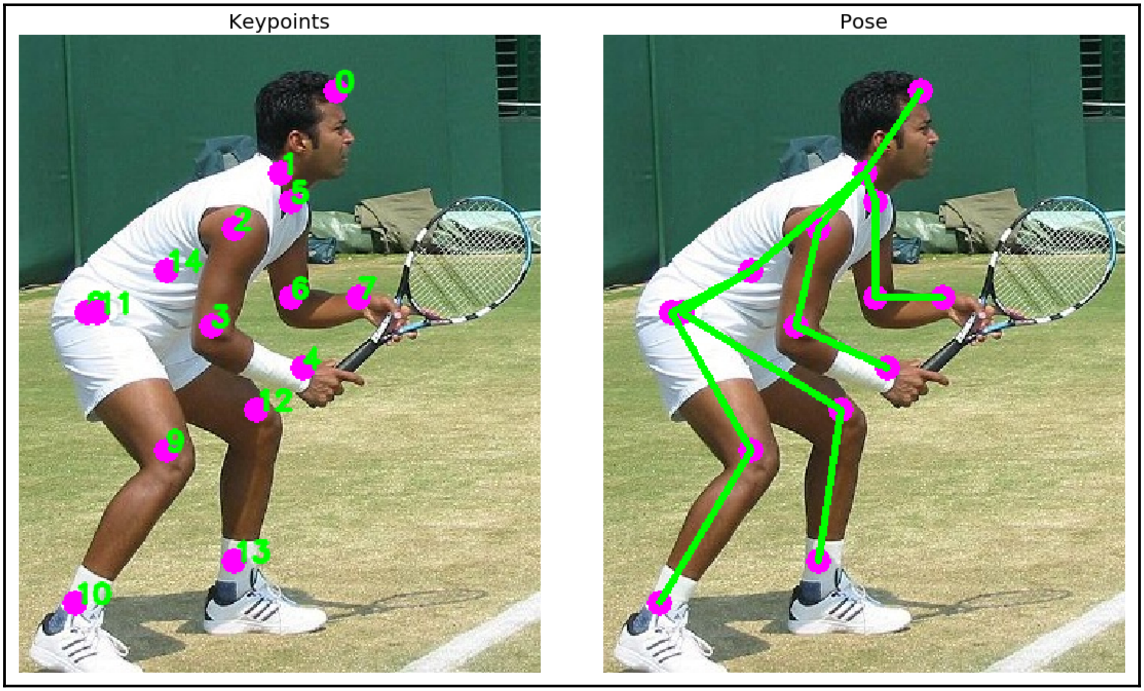

```
log loss:
training   (min:    0.000, max:    0.145, cur:    0.000)
validation (min:    0.005, max:    0.058, cur:    0.005)

accuracy:
training   (min:    0.028, max:    0.997, cur:    0.997)
validation (min:    0.035, max:    0.936, cur:    0.934)
Train Epoch: 19 [0/34799 (0%)]  train Loss: 0.002379
Train Epoch: 19 [6400/34799 (7%)]       train Loss: 0.002805
Train Epoch: 19 [12800/34799 (15%)]     train Loss: 0.003652
```


Vehicles over 3.5 metric tons prohibited
Speed limit (30km/h)
Keep right
Turn right ahead
Right-of-way at the next intersection
Keep right
General caution
Priority road
Road work
Ahead only
Priority road
Speed limit (100km/h)
Beware of ice/snow
Speed limit (100km/h)
Speed limit (70km/h)
No passing
Double curve
Dangerous curve to the right
Pedestrians
Keep right
Speed limit (70km/h)
Turn right ahead
No passing
Dangerous curve to the left
Speed limit (30km/h)
Right-of-way at the next intersection
Yield
No passing for vehicles over 3.5 metric tons
No passing for vehicles over 3.5 metric tons
Right-of-way at the next intersection
Speed limit (80km/h)
No entry
Turn left ahead
Beware of ice/snow
Speed limit (50km/h)
No entry
Speed limit (60km/h)
Priority road
Vehicles over 3.5 metric tons prohibited
Speed limit (120km/h)
Speed limit (100km/h)
Beware of ice/snow
General caution
Priority road
Road narrows on the right
Stop
Speed limit (60km/h)
No passing for vehicles over 3.5 metric tons
Wild animals crossing
Road work
Keep left
Yield
No vehicles
No passing
Yield
Ahead only
Speed limit (80km/h)
Traffic signals
No passing
Vehicles over 3.5 metric tons prohibited
Keep right
No passing for vehicles over 3.5 metric tons
Speed limit (70km/h)
No passing

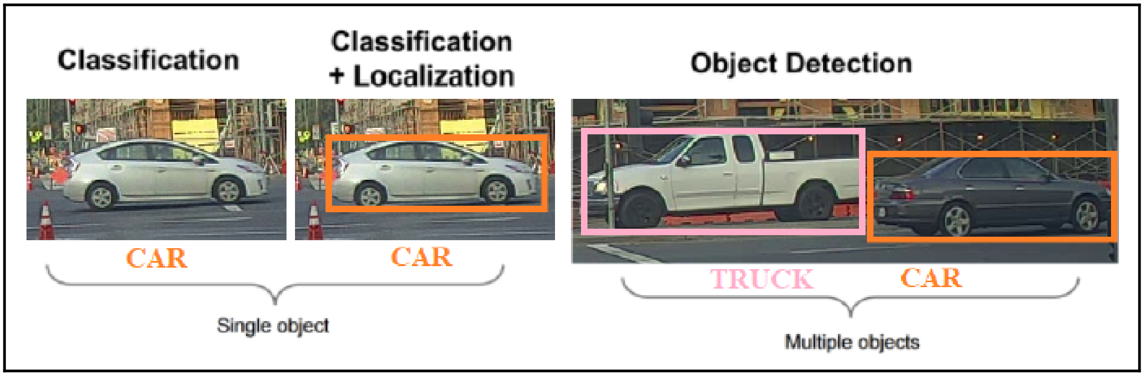


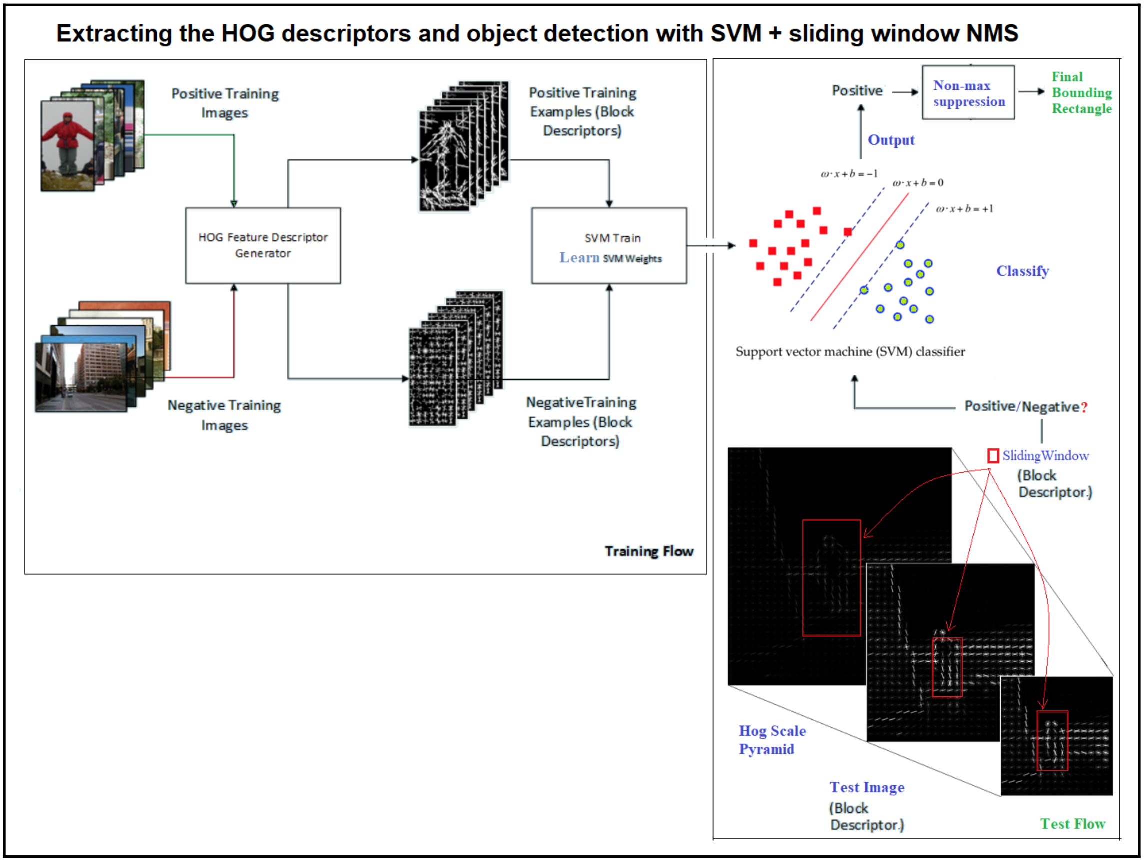
Taken from *https://arxiv.org/pdf/1611.08050.pdf*


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

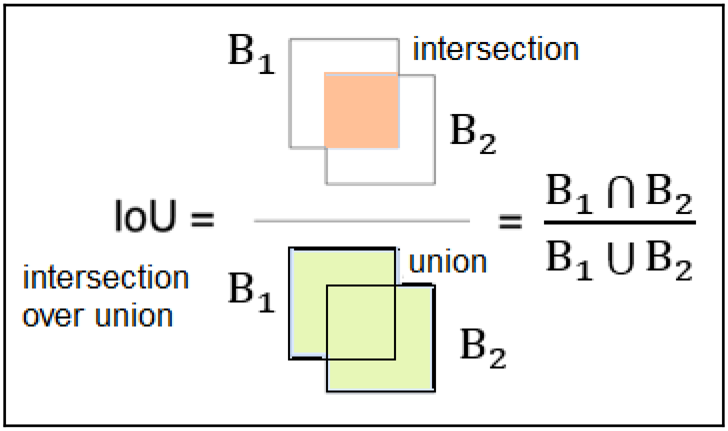

Keypoints
Pose
0
1
5
2
14
11
6
7
3
4
12
9
13
10

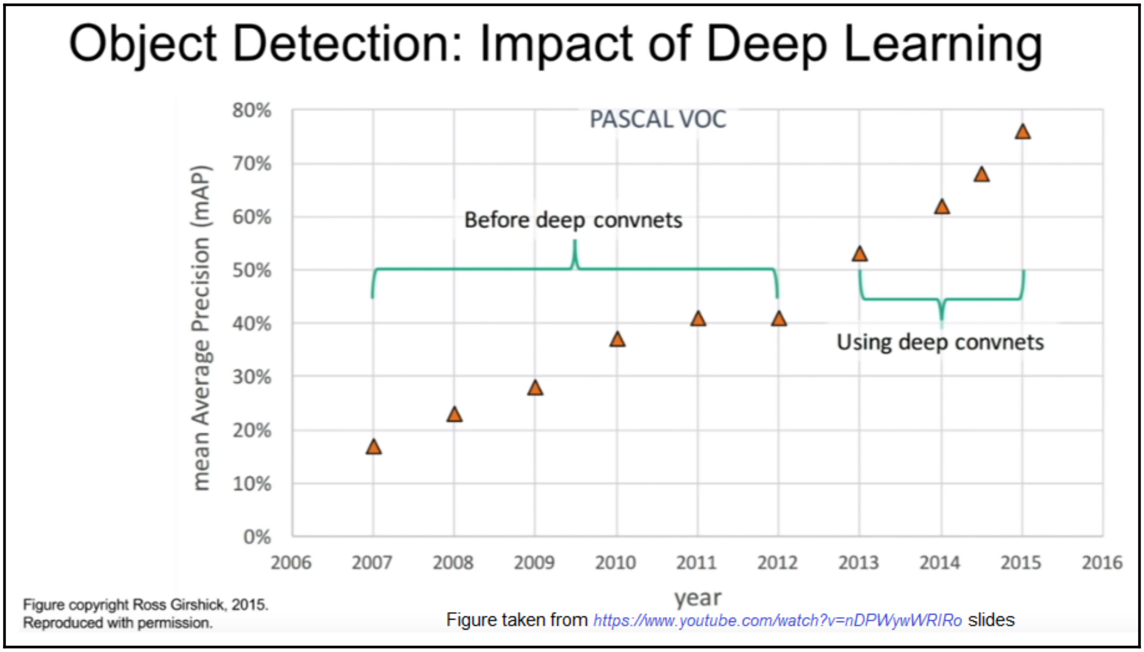
Chapter 8: Object Detection in Images

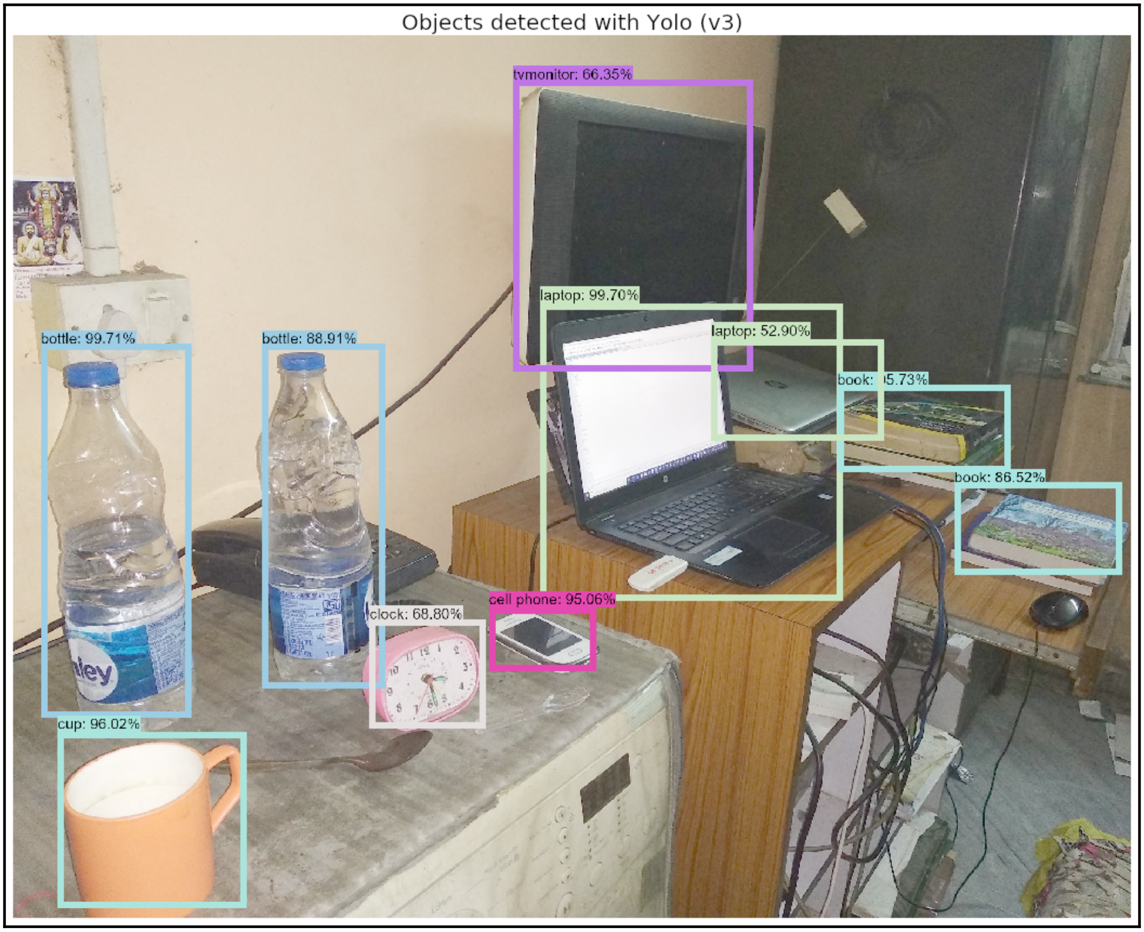


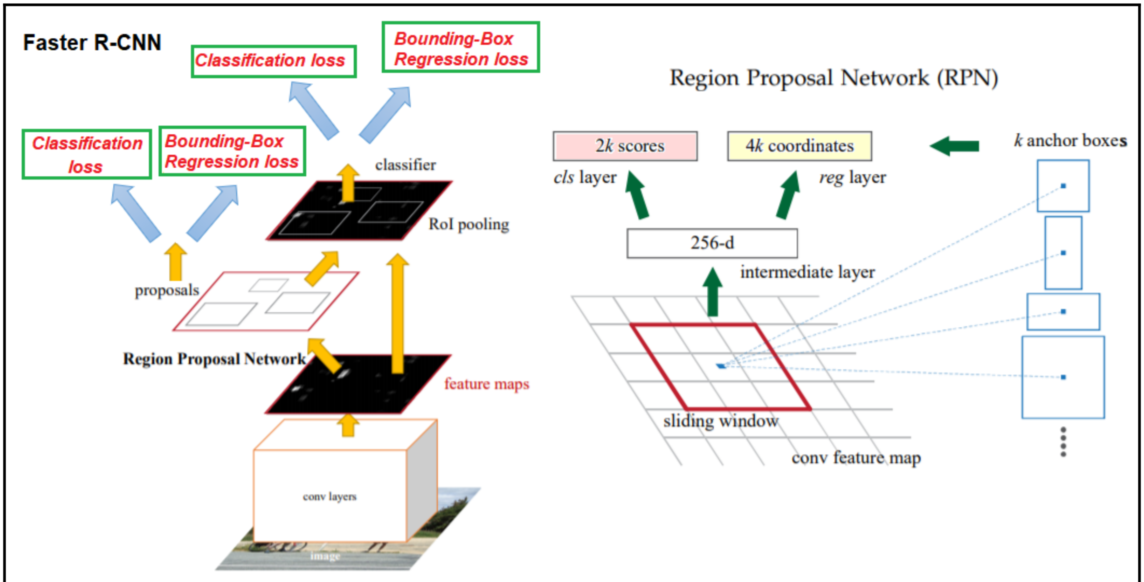



Boundingboxes found by HOG-SVM with meanshift grouping
PICNICGARDEN 39

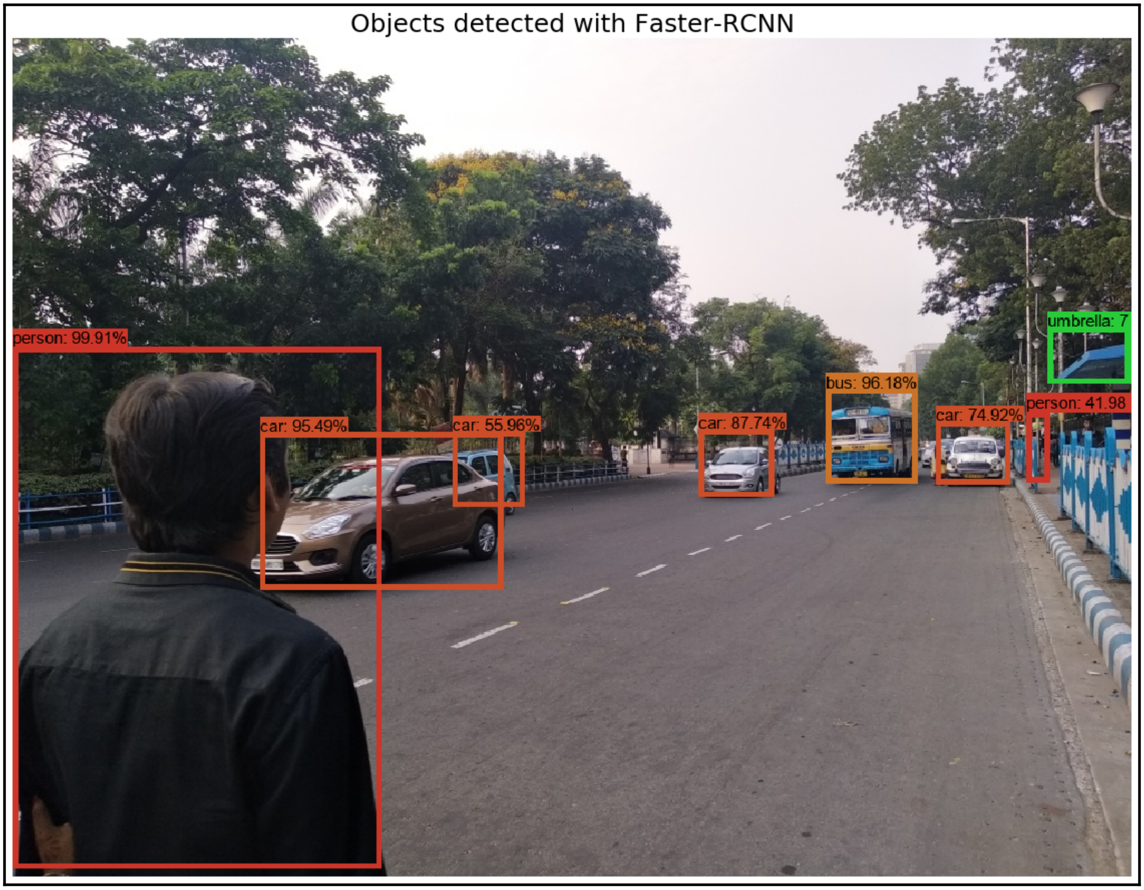

B_1
intersection
B_2
IoU =
$= \frac{B_1 \cap B_2}{B_1 \cup B_2}$
union
B_1
B_2
intersection
over union

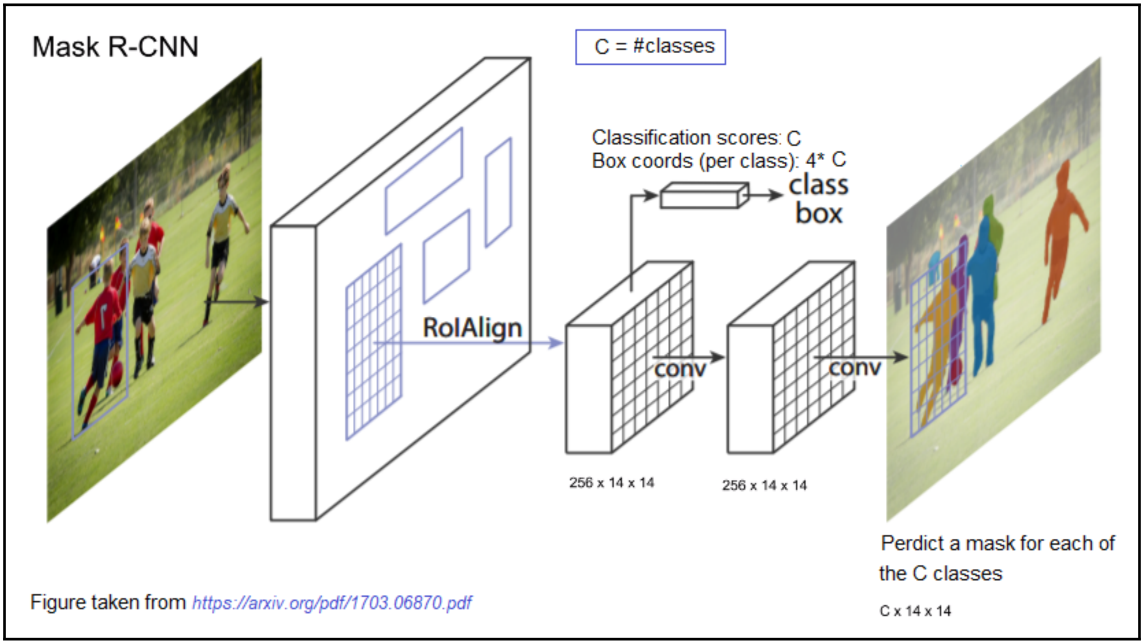

Object Detection: Impact of Deep Learning
PASCAL VOC
mean Average Precision (mAP)
80%
70%
60%
50%
40%
30%
20%
10%
0%
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
year
Before deep convnets
Using deep convnets
Figure copyright Ross Girshick, 2015.
Reproduced with permission.
Figure taken from https://www.youtube.com/watch?v=nDPWywWRIRo slides


Objects detected with Yolo (v3)
tvmonitor: 66.35%
laptop: 99.70%
laptop: 52.90%
book:
book: 86.52%
bottle: 99.71%
bottle: 88.91%
clock: 68.80%
cell phone: 95.06%
cup: 96.02%



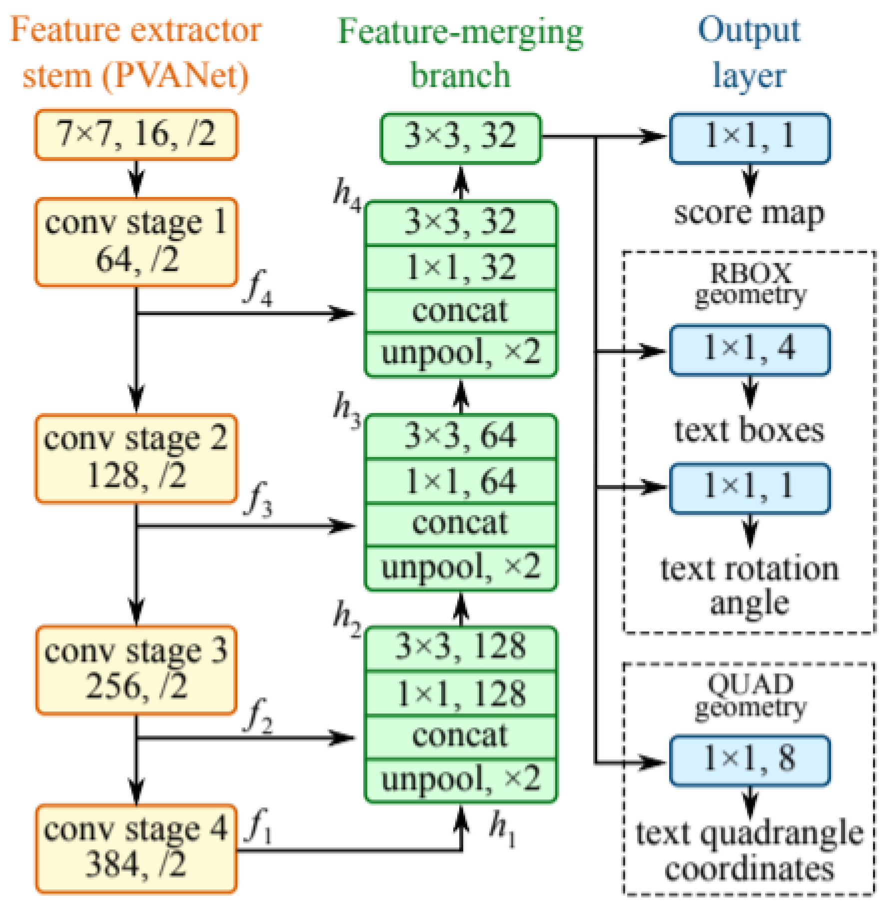
Images taken from https://arxiv.org/pdf/1506.01497.pdf


Objects detected with Faster-RCNN
person: 99.91%
car: 95.49%
car: 55.96%
car: 87.74%
bus: 96.18%
car: 74.92%
umbrella: 7
person: 41.98


Mask R-CNN
C = #classes
Classification scores: C
Box coords (per class): 4* C
class
box
RoIAlign
conv
conv
256 x 14 x 14
256 x 14 x 14
Perdict a mask for each of
the C classes
C x 14 x 14
Figure taken from https://arxiv.org/pdf/1703.06870.pdf

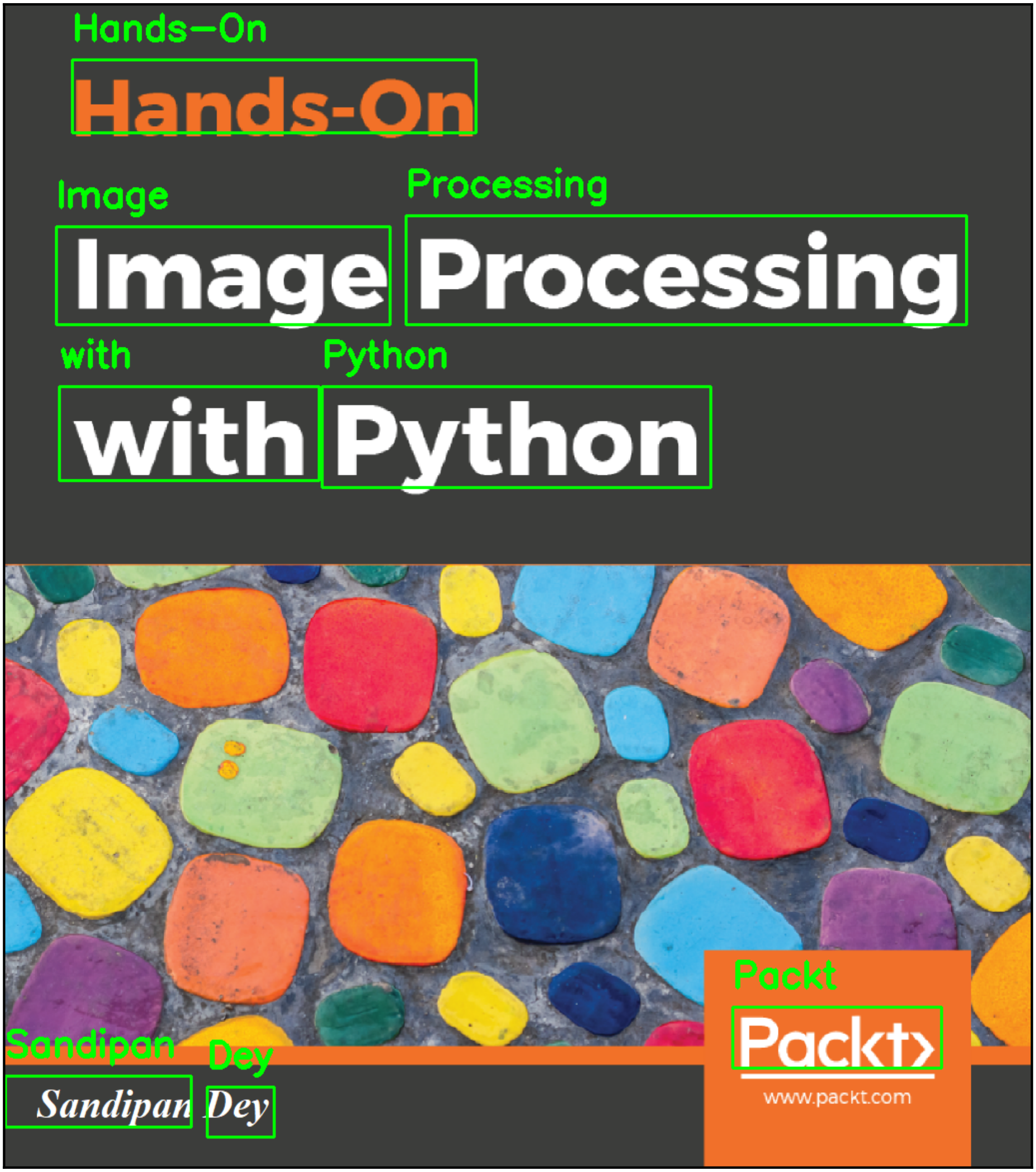

Original Image
Objects detected with Mask-RCNN
person:1.00
car:1.00
car:0.77
car:0.99
bus:0.99
car:0.773

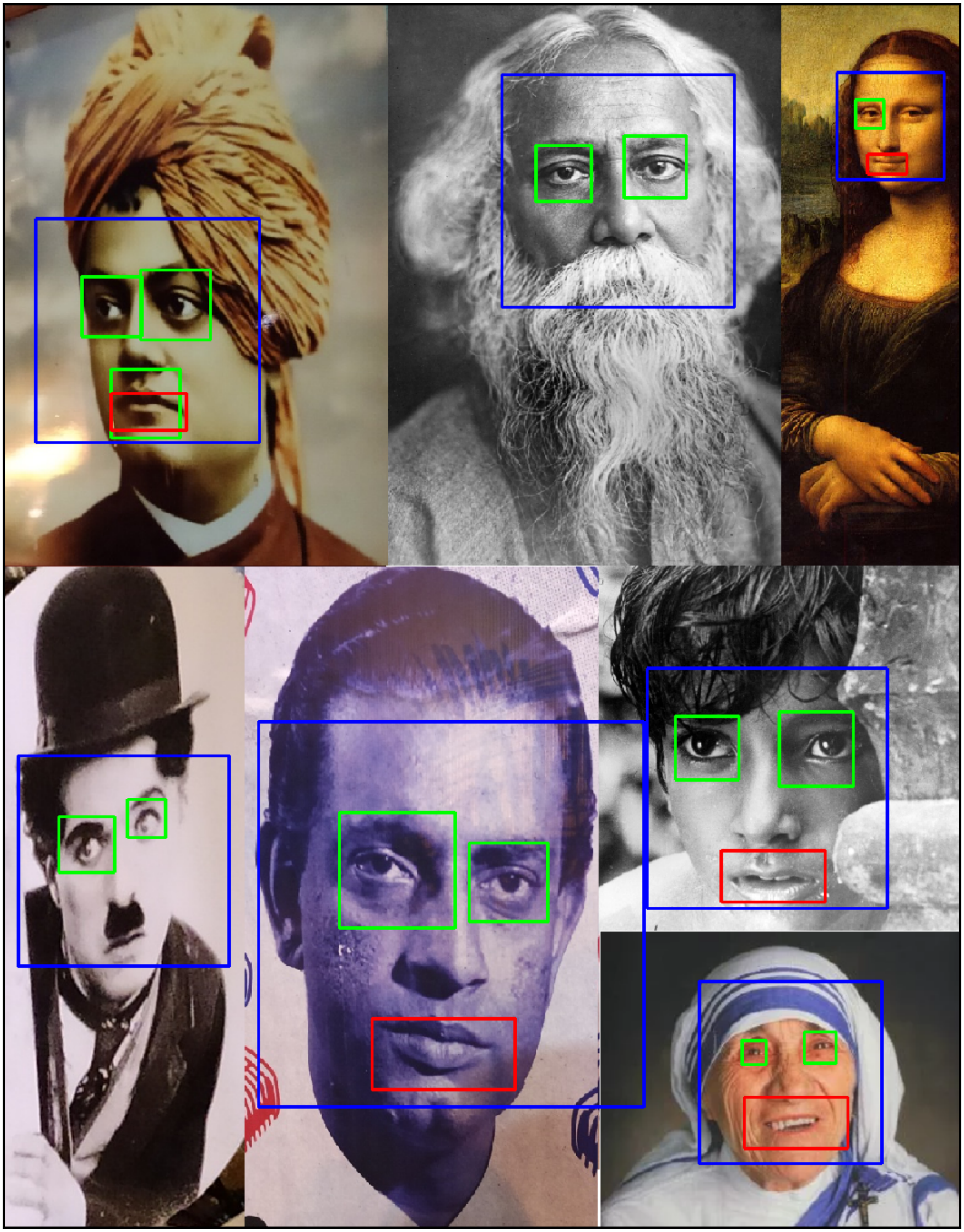

Tracking cars in a video with CSRT MultiTracker
Frame 0
Frame 4
Frame 8
Frame 12
Frame 16
Frame 20



Structure of **EAST** text detection FCN

Figure Taken from https://arxiv.org/pdf/1704.03155.pdf


Hands—On
Hands-On
Image
Processing
Image Processing
with
Python
with Python
Packt
Packt>
www.packt.com
Sandipan
Dey
Sandipan Dey



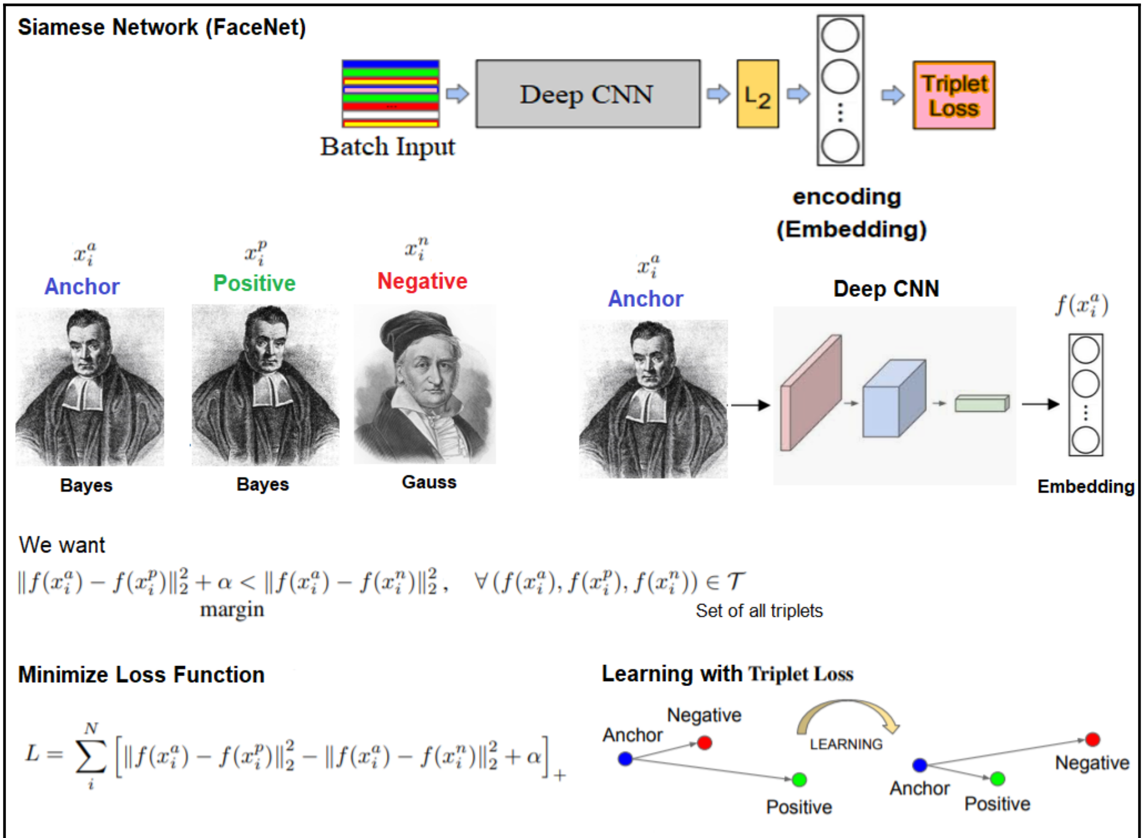
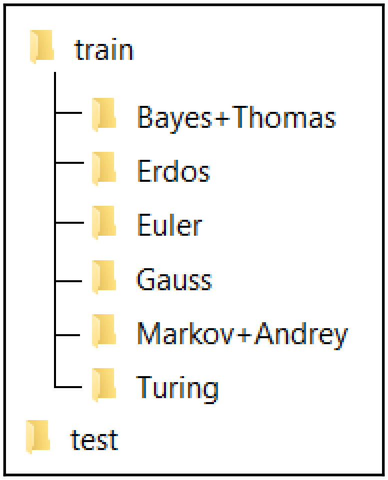
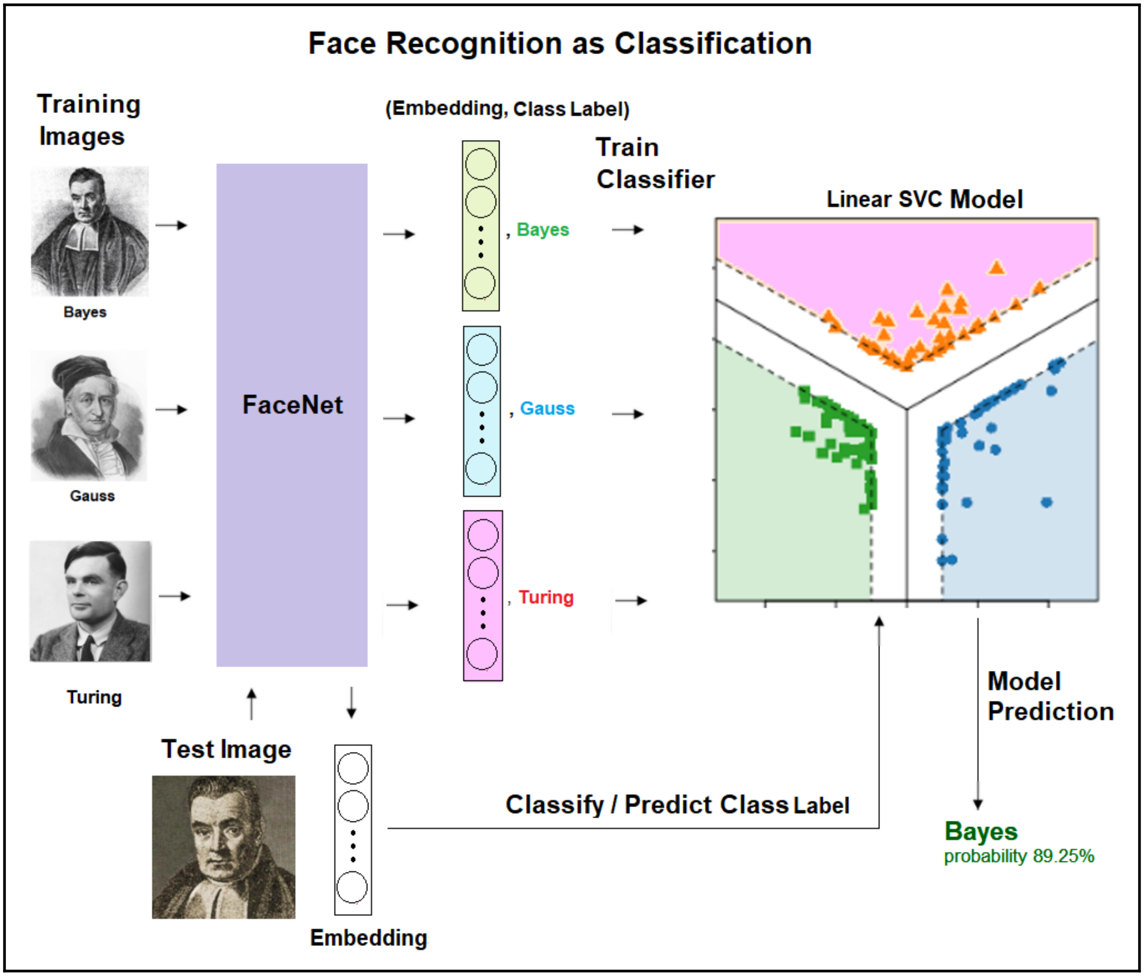


Figure adapted from https://arxiv.org/pdf/1503.03832.pdf

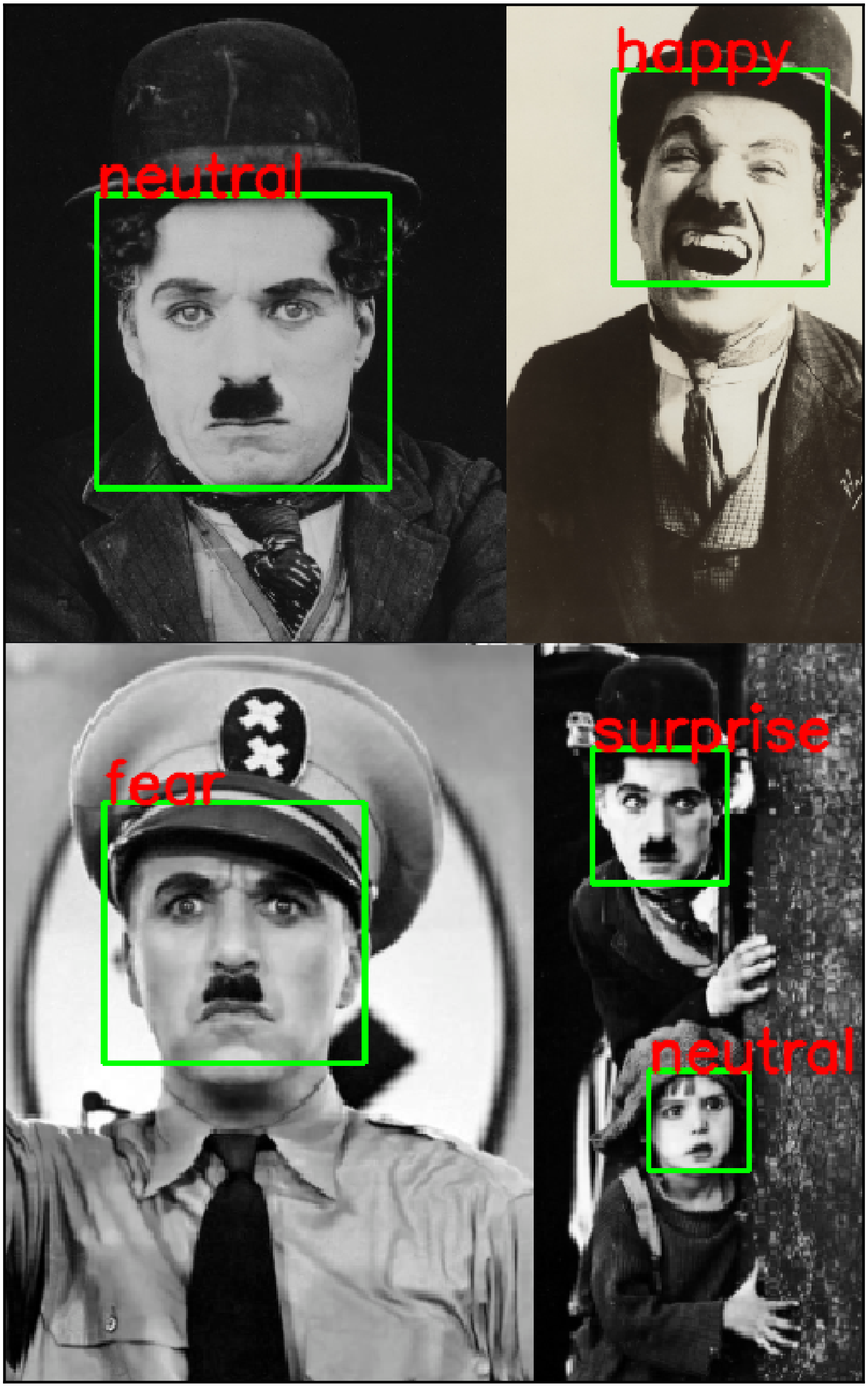

train
Bayes+Thomas
Erdos
Euler
Gauss
Markov+Andrey
Turing
test


Erdos
Erdos
Gauss
Markov+Andrey
Gauss
Turing
Bayes+Thomas
Euler
Turing
Bayes+Thomas
Bayes+Thomas
Gauss
Markov+Andrey
Gauss
Erdos
Gauss
Euler
Turing
Turing
Erdos
Euler
Gauss
Markov+Andrey
Euler
Gauss
Euler
Turing
Gauss
Bayes+Thomas
Erdos
Markov+Andrey
Turing
Turing
Turing
Markov+Andrey
Bayes+Thomas

Bayes+Thomas (80.349)


Face Recognition as Classification
Training Images
Bayes
Gauss
Turing
FaceNet
(Embedding, Class Label)
, Bayes
, Gauss
, Turing
Train Classifier
Linear SVC Model
Test Image
Embedding
Classify / Predict Class Label
Model Prediction
Bayes
probability 89.25%


bob_dylan
53, M
celine_dion
31, F
john_lennon
37, M


neutral
happy
fear
surprise
neutral

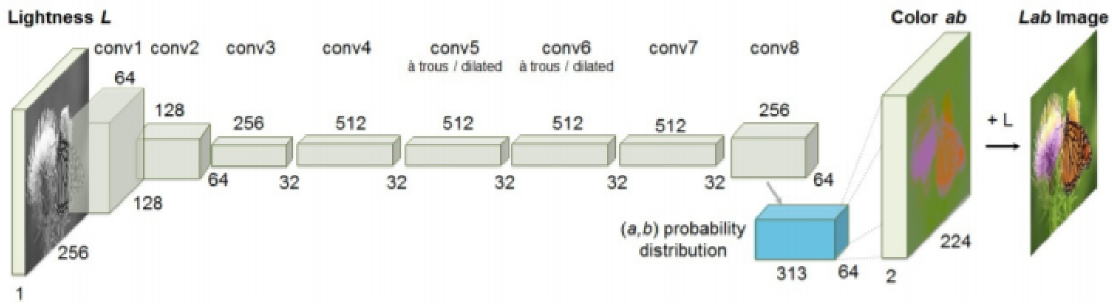
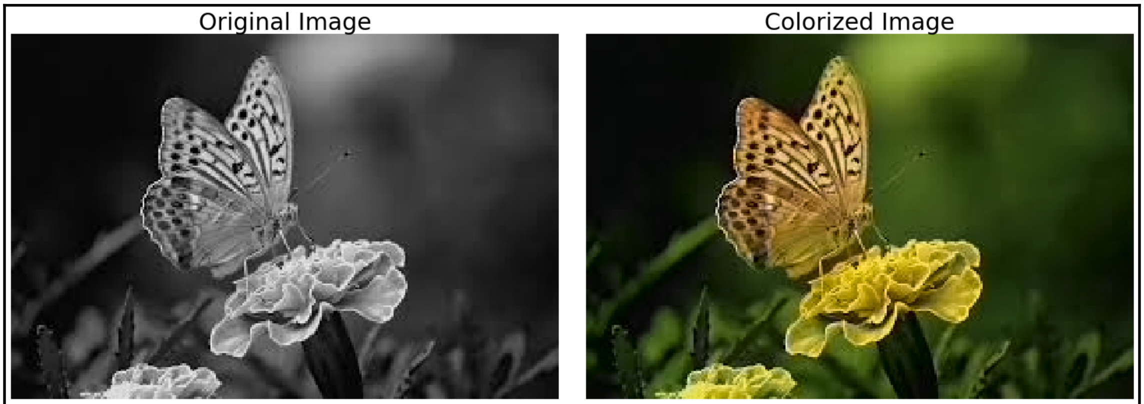
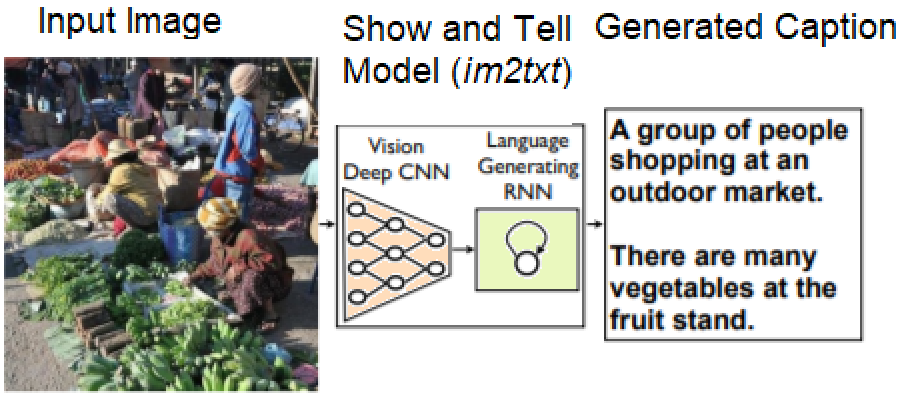
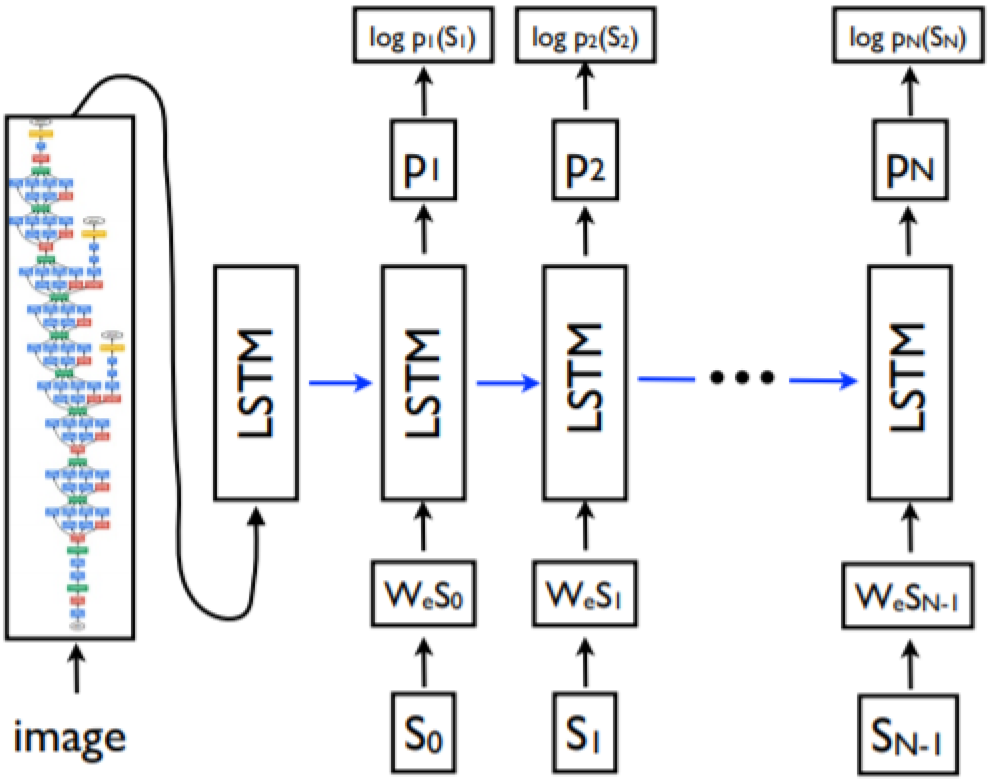


figure taken from *https://arxiv.org/pdf/1603.08511.pdf*

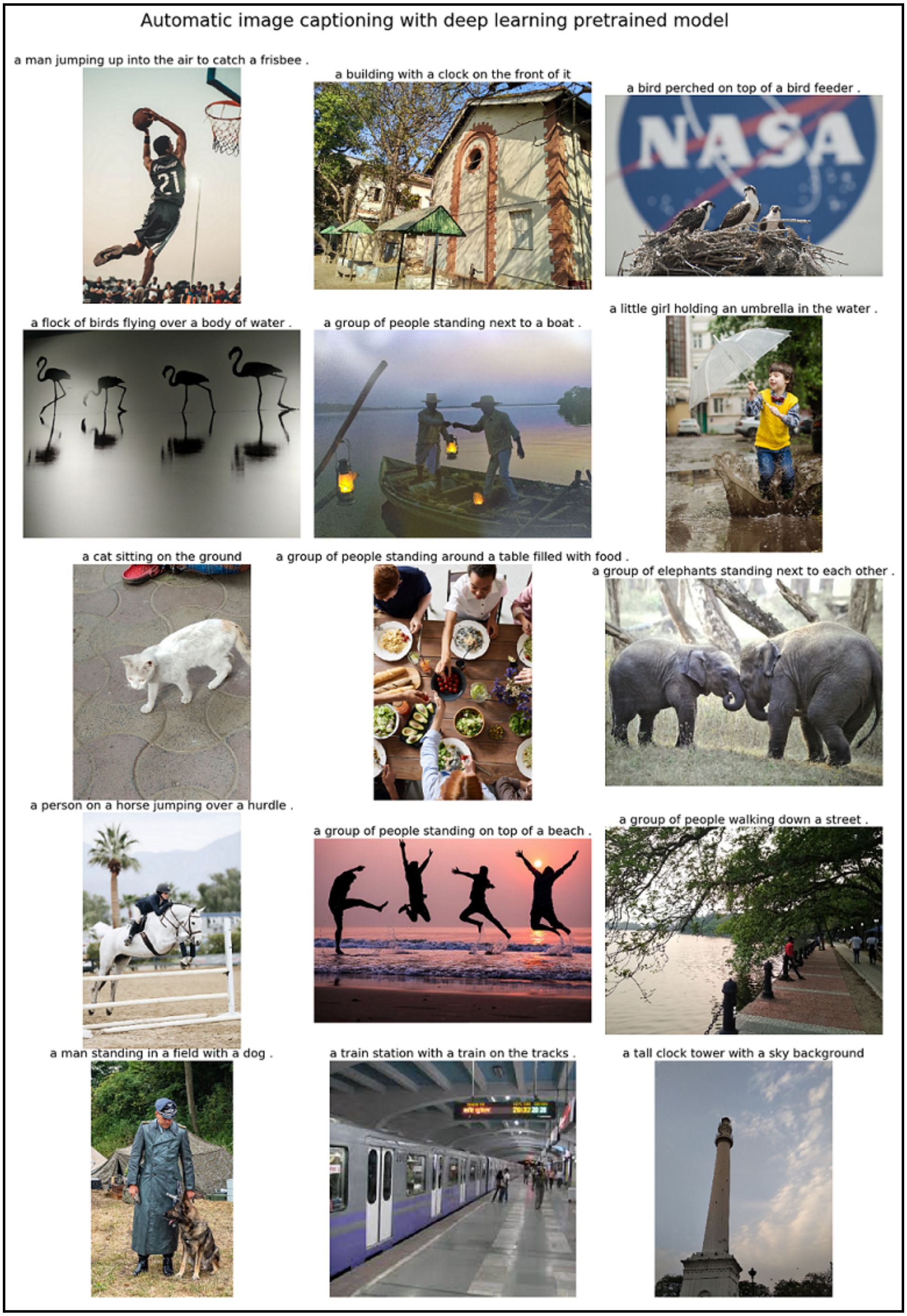


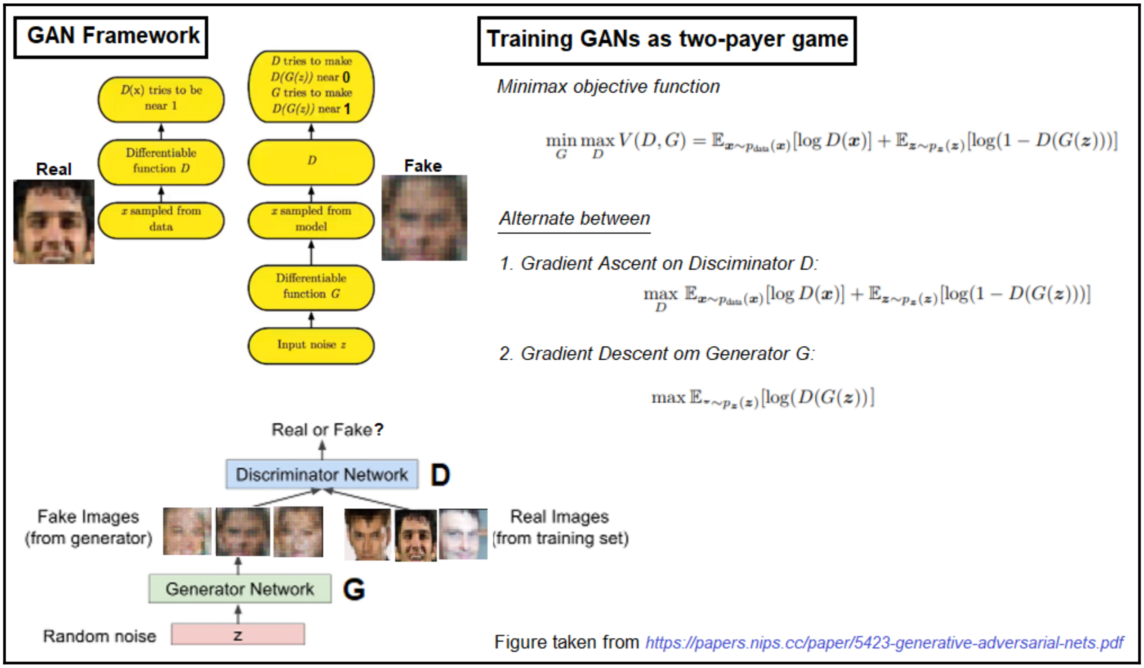


taken from https://arxiv.org/pdf/1609.06647.pdf

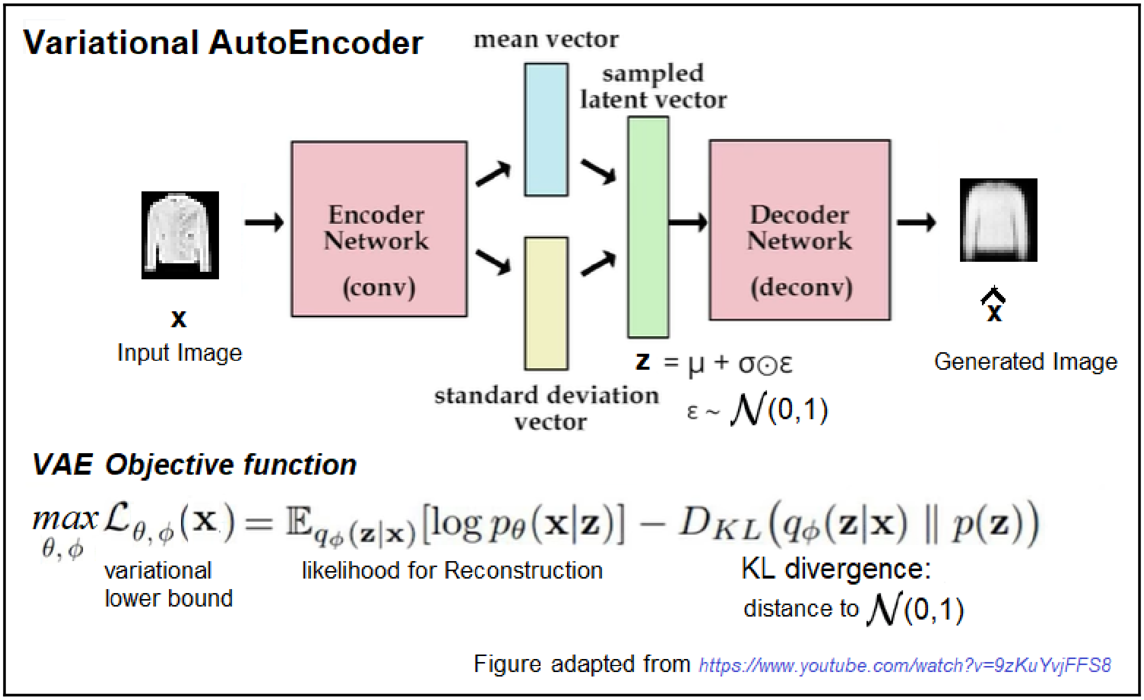


```
im2txt
├── cpt
├── data
├── inference_utils
├── ops
└── __init__.py
```

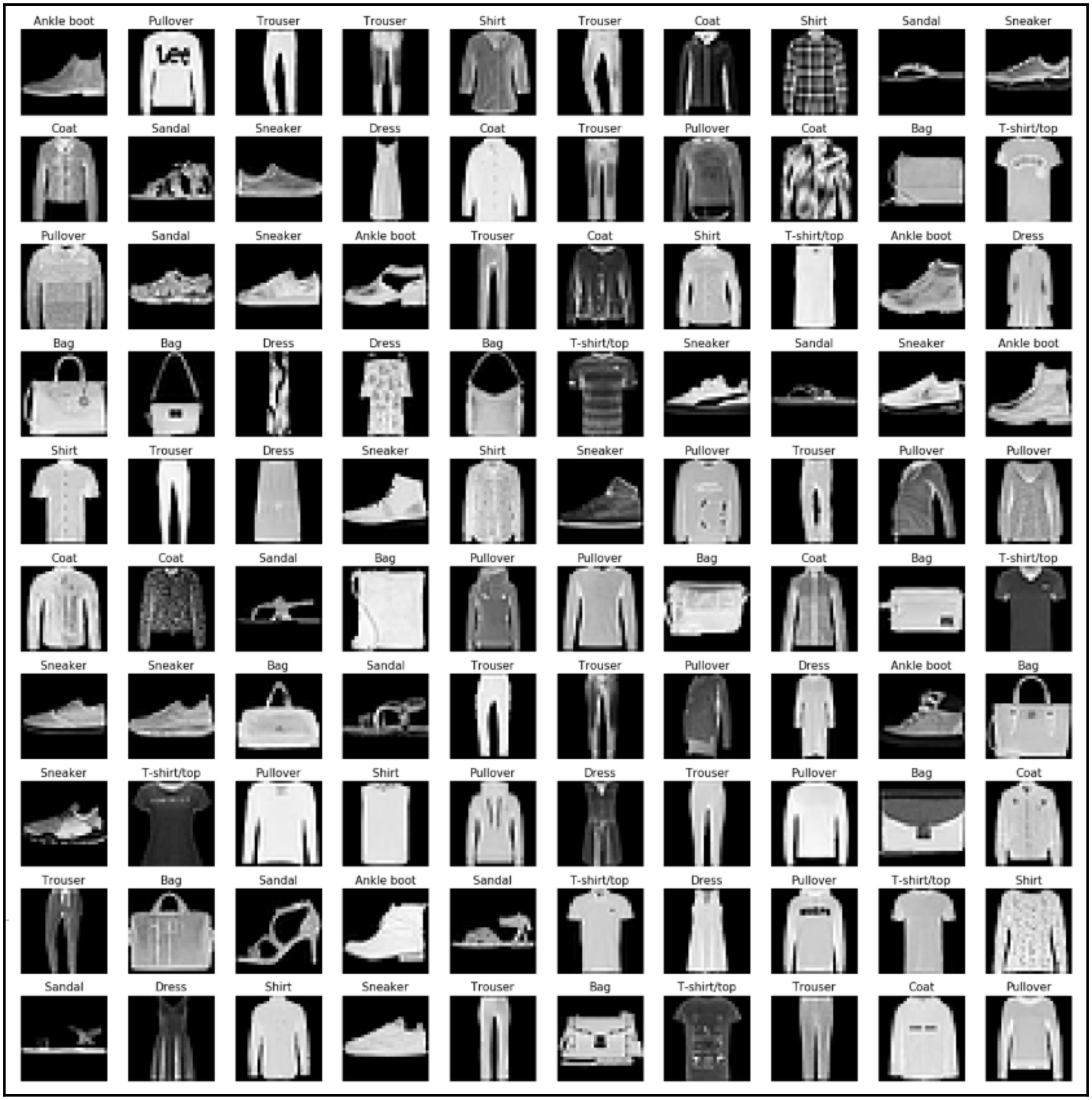

Automatic image captioning with deep learning pretrained model
a man jumping up into the air to catch a frisbee .
a building with a clock on the front of it
a bird perched on top of a bird feeder .
NASA
a flock of birds flying over a body of water .
a group of people standing next to a boat .
a little girl holding an umbrella in the water .
a cat sitting on the ground
a group of people standing around a table filled with food .
a group of elephants standing next to each other .
a person on a horse jumping over a hurdle .
a group of people standing on top of a beach .
a group of people walking down a street .
a man standing in a field with a dog .
a train station with a train on the tracks .
a tall clock tower with a sky background


GAN Framework
D(x) tries to be near 1
Differentiable function D
x sampled from data
Real
D tries to make D(G(z)) near 0
G tries to make D(G(z)) near 1
D
x sampled from model
Differentiable function G
Input noise z
Fake
Real or Fake?
Discriminator Network
D
Fake Images (from generator)
Real Images (from training set)
Generator Network
G
Random noise
z
Training GANs as two-payer game
Minimax objective function
$\min_G \max_D V(D,G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$
Alternate between
1. Gradient Ascent on Discriminator D:
$\max_D \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$
2. Gradient Descent om Generator G:
$\max \mathbb{E}_{z \sim p_z(z)}[\log(D(G(z)))]$
Figure taken from https://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

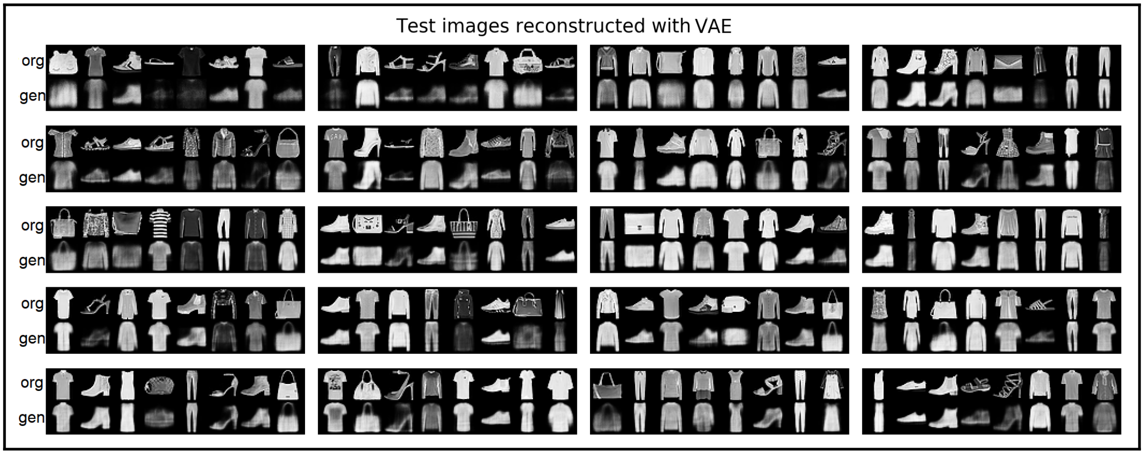

Images generated by GAN (epoch = 10)

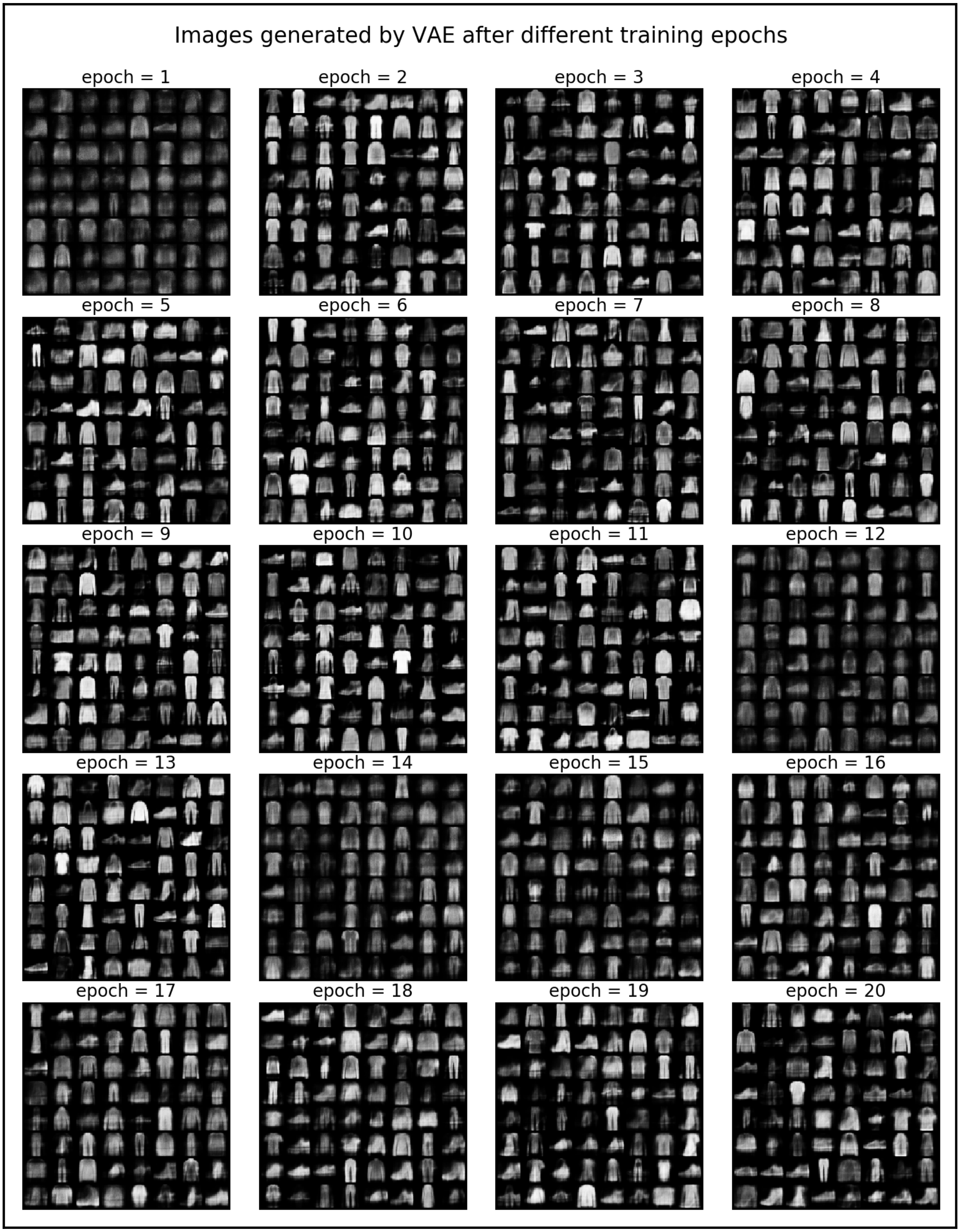


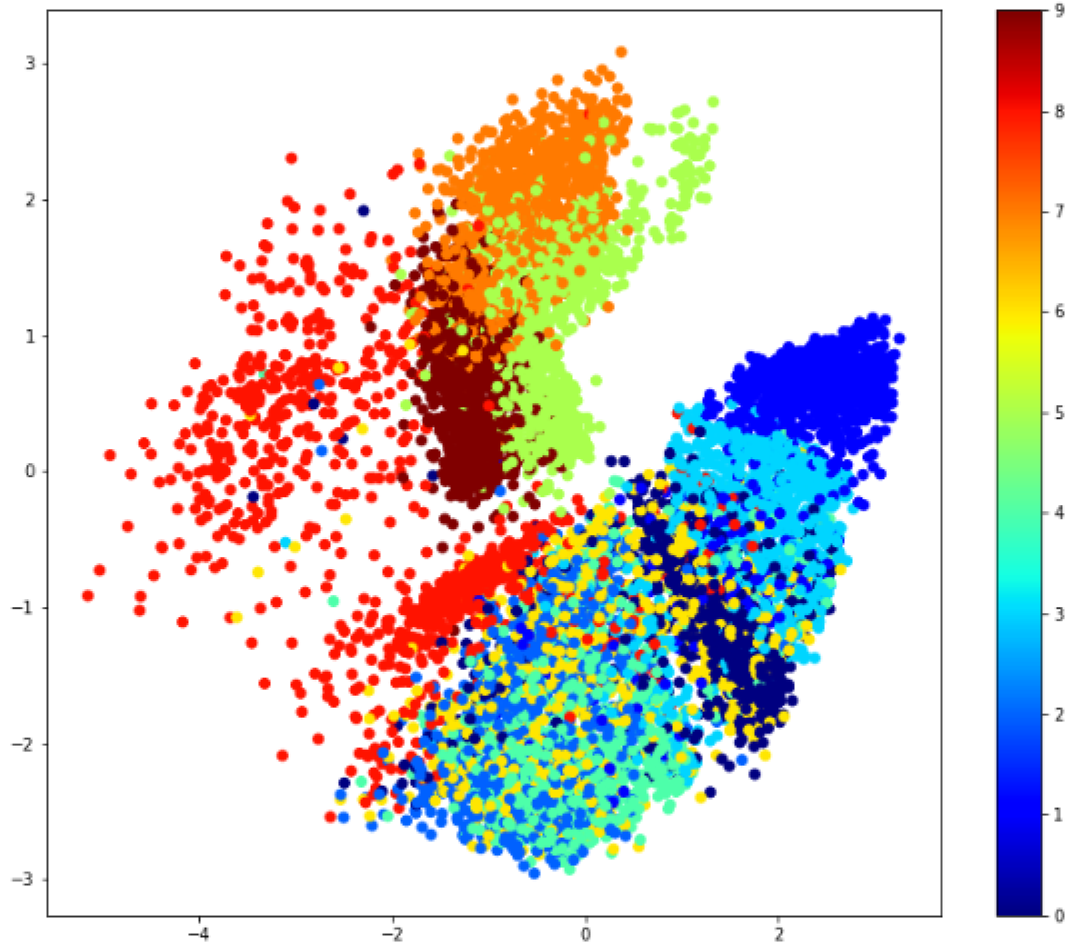
Variational AutoEncoder
mean vector
sampled latent vector
Encoder Network (conv)
Decoder Network (deconv)
x
Input Image
Generated Image
z = μ + σ⊙ε
ε ~ N(0,1)
standard deviation vector
VAE Objective function
$\max_{\theta,\phi} \mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z}))$
variational lower bound
likelihood for Reconstruction
KL divergence:
distance to N(0,1)
Figure adapted from https://www.youtube.com/watch?v=9zKuYvjFFS8



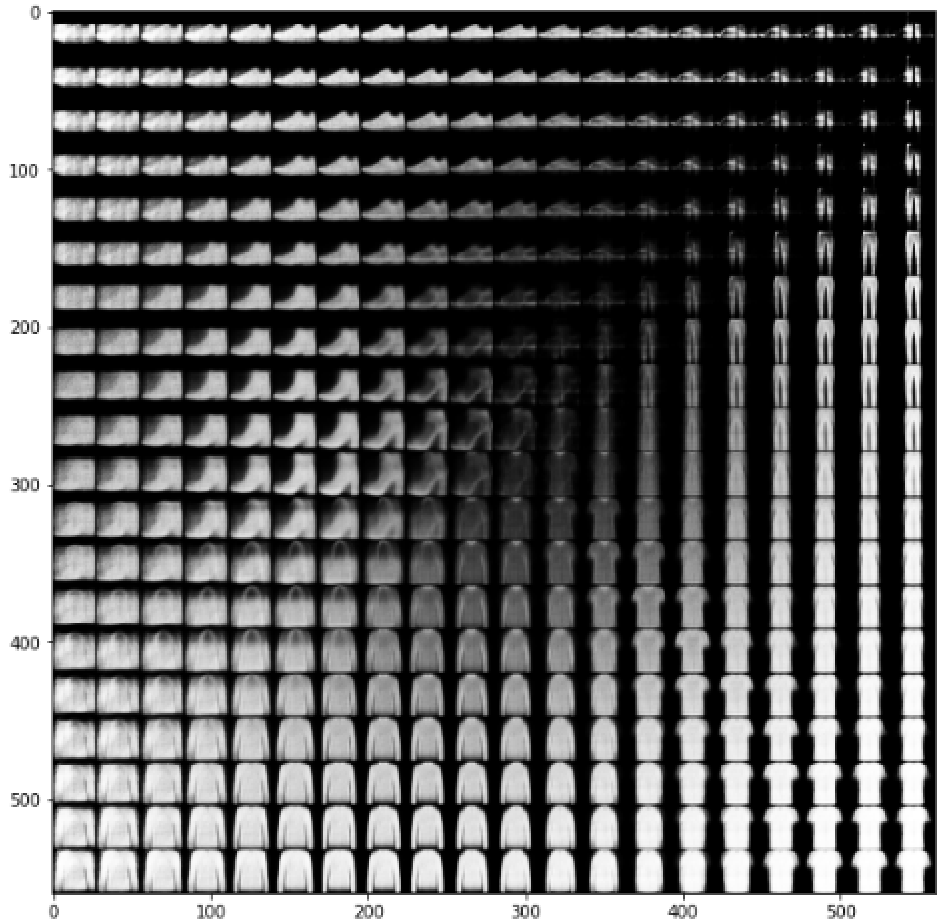
Ankle boot
Pullover
Trouser
Trouser
Shirt
Trouser
Coat
Shirt
Sandal
Sneaker
Coat
Sandal
Sneaker
Dress
Coat
Trouser
Pullover
Coat
Bag
T-shirt/top
Pullover
Sandal
Sneaker
Ankle boot
Trouser
Coat
Shirt
T-shirt/top
Ankle boot
Dress
Bag
Bag
Dress
Dress
Bag
T-shirt/top
Sneaker
Sandal
Sneaker
Ankle boot
Shirt
Trouser
Dress
Sneaker
Shirt
Sneaker
Pullover
Trouser
Pullover
Pullover
Coat
Coat
Sandal
Bag
Pullover
Pullover
Bag
Coat
Bag
T-shirt/top
Sneaker
Sneaker
Bag
Sandal
Trouser
Trouser
Pullover
Dress
Ankle boot
Bag
Sneaker
T-shirt/top
Pullover
Shirt
Pullover
Dress
Trouser
Pullover
Bag
Coat
Trouser
Bag
Sandal
Ankle boot
Sandal
T-shirt/top
Dress
Pullover
T-shirt/top
Shirt
Sandal
Dress
Shirt
Sneaker
Trouser
Bag
T-shirt/top
Trouser
Coat
Pullover

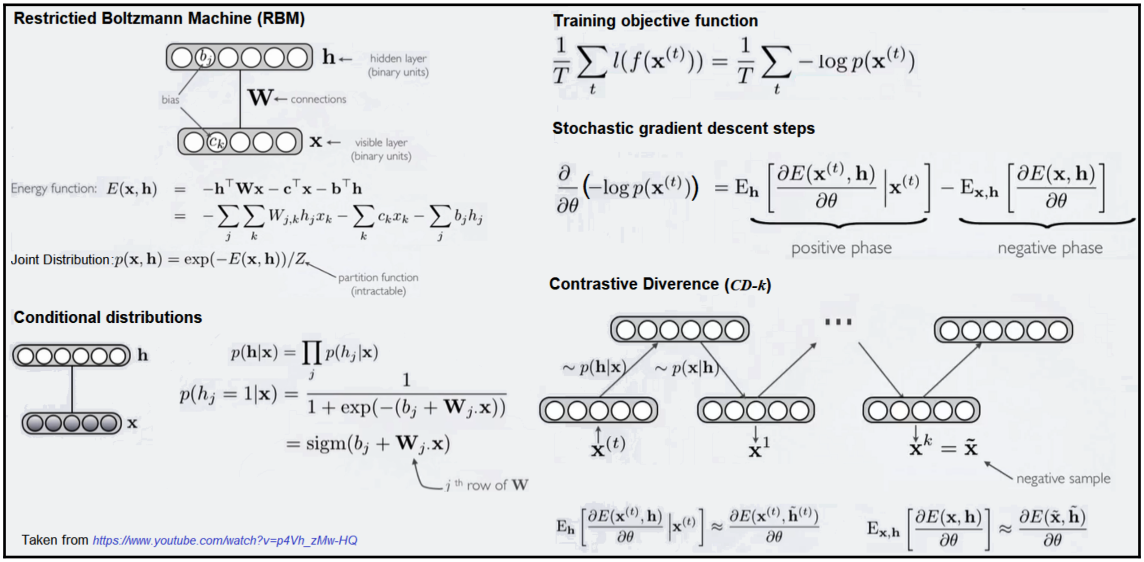

Test images reconstructed with VAE
org
gen

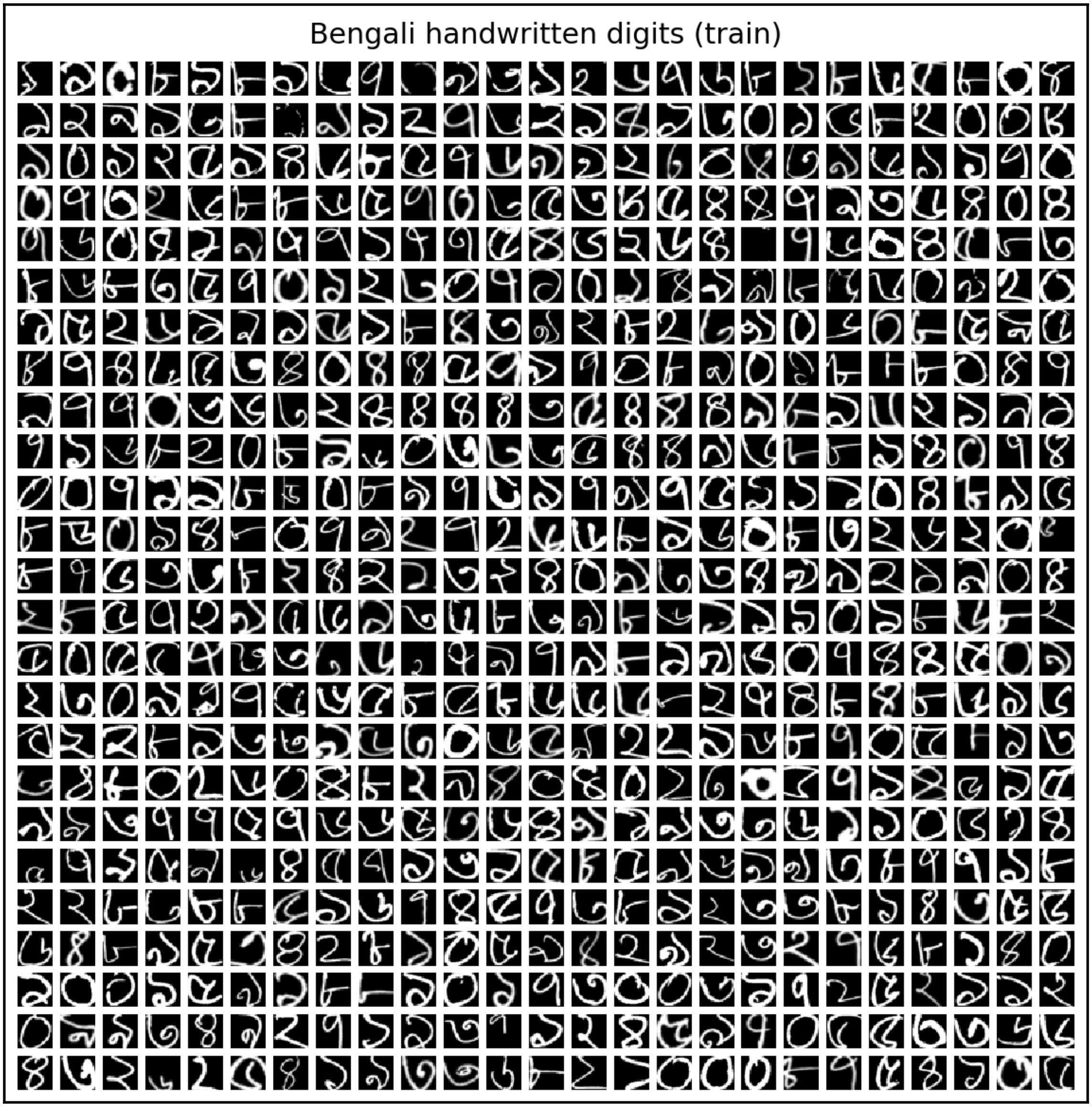

Images generated by VAE after different training epochs
epoch = 1
epoch = 2
epoch = 3
epoch = 4
epoch = 5
epoch = 6
epoch = 7
epoch = 8
epoch = 9
epoch = 10
epoch = 11
epoch = 12
epoch = 13
epoch = 14
epoch = 15
epoch = 16
epoch = 17
epoch = 18
epoch = 19
epoch = 20

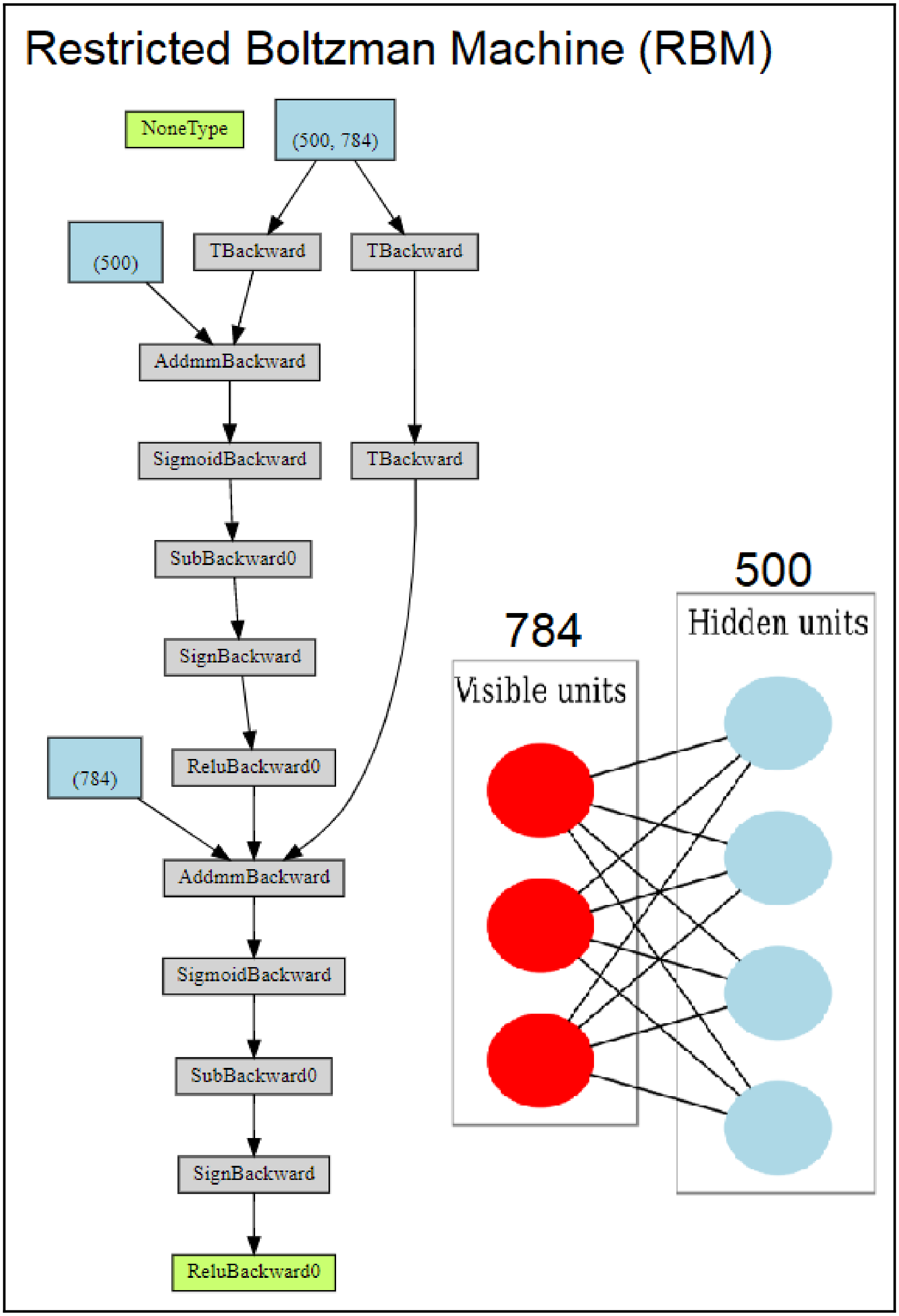


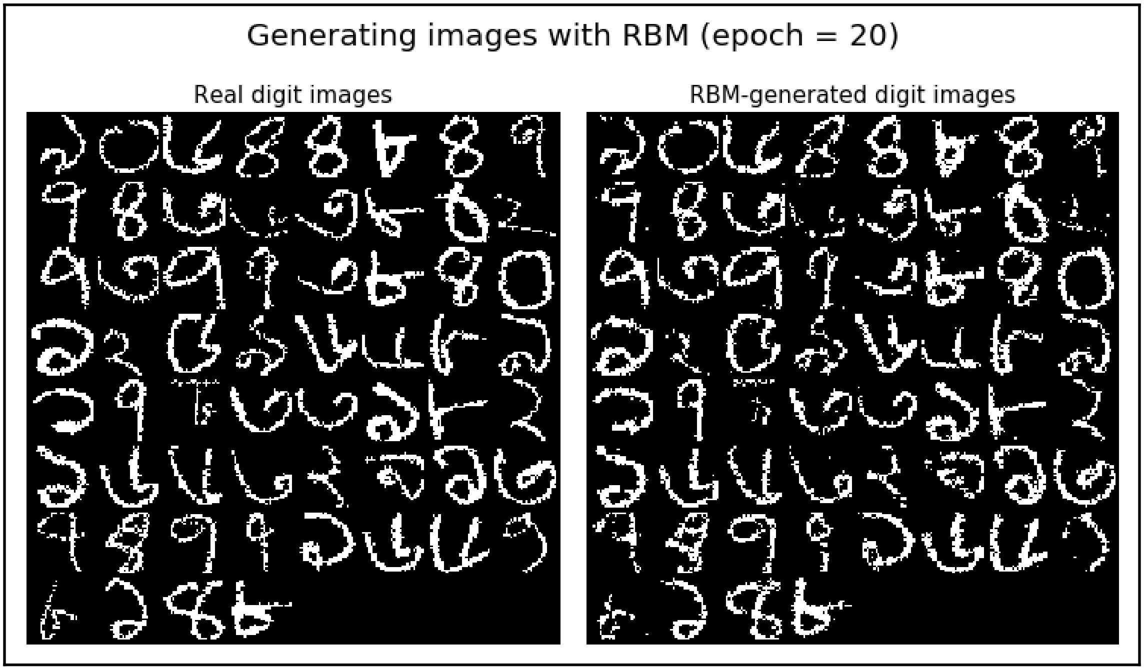

```
{0: 'T-shirt/top',
 1: 'Trouser',
 2: 'Pullover',
 3: 'Dress',
 4: 'Coat',
 5: 'Sandal',
 6: 'Shirt',
 7: 'Sneaker',
 8: 'Bag',
 9: 'Ankle boot'}
```

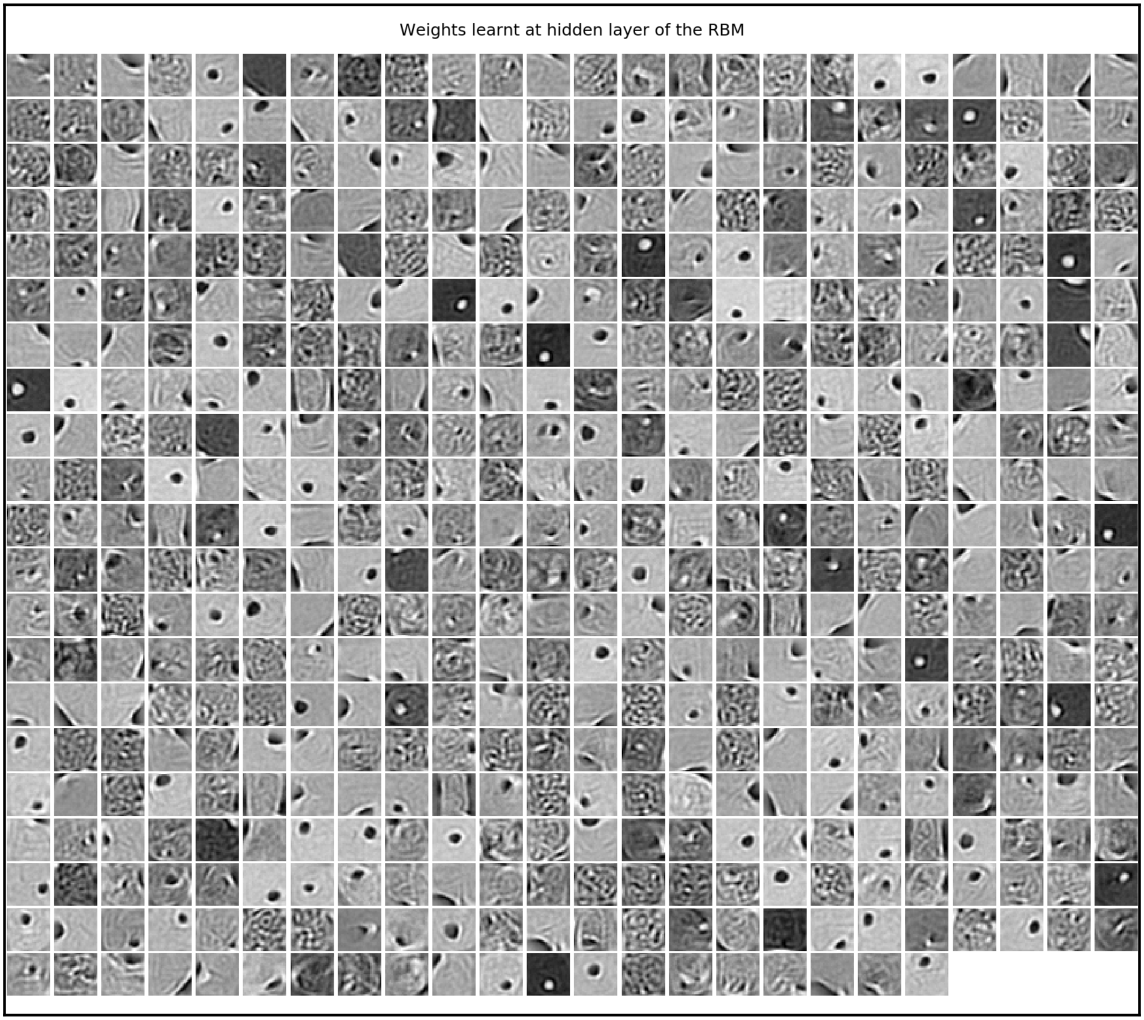

0
100
200
300
400
500
0
100
200
300
400
500


Restrictied Boltzmann Machine (RBM)
hidden layer (binary units)
bias
connections
visible layer (binary units)
Energy function: $E(\mathbf{x},\mathbf{h}) = -\mathbf{h}^\top \mathbf{W}\mathbf{x} - \mathbf{c}^\top \mathbf{x} - \mathbf{b}^\top \mathbf{h} = -\sum_j \sum_k W_{j,k} h_j x_k - \sum_k c_k x_k - \sum_j b_j h_j$
Joint Distribution: $p(\mathbf{x},\mathbf{h}) = \exp(-E(\mathbf{x},\mathbf{h}))/Z$
partition function (intractable)
Conditional distributions
$p(\mathbf{h}|\mathbf{x}) = \prod_j p(h_j|\mathbf{x})$
$p(h_j = 1|\mathbf{x}) = \frac{1}{1+\exp(-(b_j + \mathbf{W}_{j\cdot}\mathbf{x}))} = \mathrm{sigm}(b_j + \mathbf{W}_{j\cdot}\mathbf{x})$
j th row of $\mathbf{W}$
Taken from https://www.youtube.com/watch?v=p4Vh_zMw-HQ
Training objective function
$\frac{1}{T}\sum_t l(f(\mathbf{x}^{(t)})) = \frac{1}{T}\sum_t -\log p(\mathbf{x}^{(t)})$
Stochastic gradient descent steps
$\frac{\partial}{\partial\theta}\left(-\log p(\mathbf{x}^{(t)})\right) = \underbrace{\mathrm{E}_{\mathbf{h}}\left[\frac{\partial E(\mathbf{x}^{(t)},\mathbf{h})}{\partial\theta}\middle|\mathbf{x}^{(t)}\right]}_{\text{positive phase}} - \underbrace{\mathrm{E}_{\mathbf{x},\mathbf{h}}\left[\frac{\partial E(\mathbf{x},\mathbf{h})}{\partial\theta}\right]}_{\text{negative phase}}$
Contrastive Diverence (CD-k)
$\sim p(\mathbf{h}|\mathbf{x})$
$\sim p(\mathbf{x}|\mathbf{h})$
$\mathbf{x}^{(t)}$
$\mathbf{x}^{1}$
$\mathbf{x}^{k} = \tilde{\mathbf{x}}$
negative sample
$\mathrm{E}_{\mathbf{h}}\left[\frac{\partial E(\mathbf{x}^{(t)},\mathbf{h})}{\partial\theta}\middle|\mathbf{x}^{(t)}\right] \approx \frac{\partial E(\mathbf{x}^{(t)},\tilde{\mathbf{h}}^{(t)})}{\partial\theta}$
$\mathrm{E}_{\mathbf{x},\mathbf{h}}\left[\frac{\partial E(\mathbf{x},\mathbf{h})}{\partial\theta}\right] \approx \frac{\partial E(\tilde{\mathbf{x}},\tilde{\mathbf{h}})}{\partial\theta}$




Restricted Boltzman Machine (RBM)
NoneType
(500, 784)
(500)
TBackward
TBackward
AddmmBackward
SigmoidBackward
TBackward
SubBackward0
SignBackward
(784)
ReluBackward0
AddmmBackward
SigmoidBackward
SubBackward0
SignBackward
ReluBackward0
784
Visible units
500
Hidden units


Generating images with RBM (epoch = 20)
Real digit images
RBM-generated digit images


Weights learnt at hidden layer of the RBM