Morphological Network:
With applications to Computer Vision

Bhabatosh Chanda

Indian Statistical Institute, Kolkata

Acknowledgement

- Ranjan Mandal
- Sanchayan Santra

Outline

- Digital image

Outline

- Digital image
- Mathematical Morphology: Basics

Outline

- Digital image
- Mathematical Morphology: Basics
- Brief introdiction to ANN-CNN

Outline

- Digital image
- Mathematical Morphology: Basics
- Brief introdiction to ANN-CNN
- Automatic Learning of Structuring Element
    - De-raining problem

Outline

- Digital image
- Mathematical Morphology: Basics
- Brief introdiction to ANN-CNN
- Automatic Learning of Structuring Element
    - De-raining problem
- Opening-Closing Network
    - De-raining problem
    - De-hazing problem

Outline

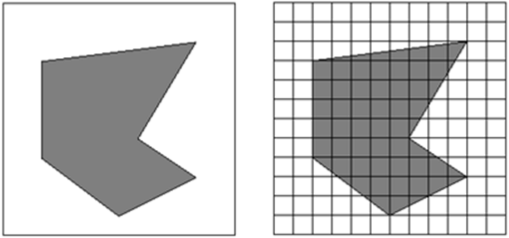
- Digital image
- Mathematical Morphology: Basics
- Brief introdiction to ANN-CNN
- Automatic Learning of Structuring Element
    - De-raining problem
- Opening-Closing Network
    - De-raining problem
    - De-hazing problem
- Conclusion

Digital image

Digital binary image: example

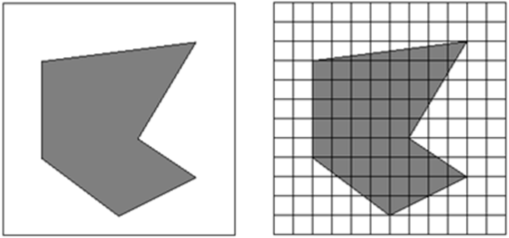
Pixel is the smallest rectangular area under the grid. It is smallest unit of a digital image that can be accessed or processed. Pixel values are either 0 or 1.

Digital binary image: example



Pixel is the smallest rectangular area under the grid. It is smallest unit of a digital image that can be accessed or processed. Pixel values are either 0 or 1.

Digital binary image: example



0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	1	1	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Pixel is the smallest rectangular area under the grid. It is smallest unit of a digital image that can be accessed or processed. Pixel values are either 0 or 1.

Digital grayscale image: example

109	106	106	110	111	106	60	98	142	114	94	98	102	120	103	109
115	126	126	137	137	136	106	146	157	146	124	128	98	96	89	82
137	118	134	145	151	147	142	157	154	140	120	105	85	77	67	63
189	150	123	131	145	143	143	134	123	110	90	65	65	67	83	102
193	191	157	101	97	104	105	101	95	98	84	94	107	119	138	155
193	189	185	168	103	74	90	106	112	125	133	136	143	154	165	170
185	187	180	179	174	142	116	128	141	148	154	160	163	164	168	171
181	185	178	173	179	171	162	153	153	160	162	164	164	163	165	167
183	184	175	149	177	179	165	164	155	153	159	164	160	159	163	164
185	182	161	126	170	188	181	167	158	153	151	155	155	158	160	163
185	178	147	125	171	183	192	181	167	157	155	160	152	156	160	160
178	179	145	139	163	171	190	195	176	164	161	162	162	150	154	155
175	177	148	150	163	169	186	197	191	169	167	167	160	158	150	152
164	169	142	154	171	172	182	190	189	174	166	168	169	165	162	150
163	165	139	153	175	163	179	189	187	178	166	167	170	169	170	168
162	161	136	144	164	163	171	189	176	179	169	169	169	164	163	169

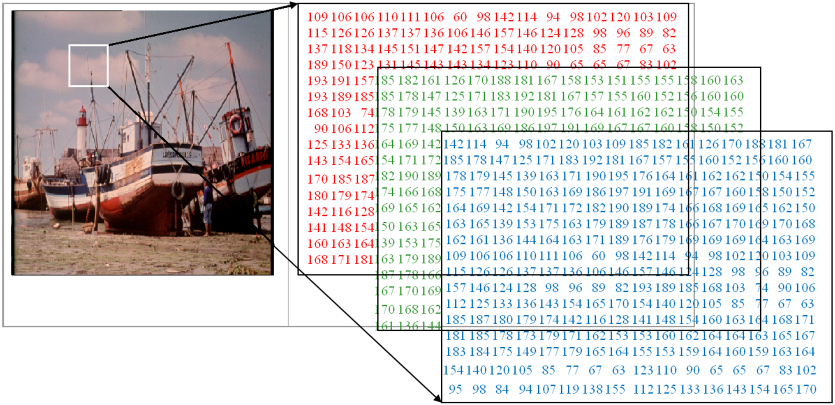
Digital image representation

Digital image (grayscale or binary) can be defined over a two-dimensional space $(i, j) \in \mathcal{Z}^2$ such that

$$[g(i,j)] = \begin{bmatrix} g(0,0) & g(0,1) & \cdots & g(0,N-1) \\ g(1,0) & g(1,1) & \cdots & g(1,N-1) \\ g(2,0) & g(2,1) & \cdots & g(2,N-1) \\ \vdots & \vdots & \cdots & \vdots \\ g(M-1,0) & g(M-1,1) & \cdots & g(M-1,N-1) \end{bmatrix}$$

where $g(i,j) \in \{0, \ldots, L-1\}$ are non-negative integers.
For binary image $L = 2$ and for graylevel image $L = 255$

Digital colour image: example



Digital colour image representation

Digital colour image can be defined over a two-dimensional space $(i,j) \in \mathcal{Z}^2$ such that

$$[g_c(i,j)] = \begin{bmatrix} g_c(0,0) & g_c(0,1) & \cdots & g_c(0,N-1) \\ g_c(1,0) & g_c(1,1) & \cdots & g_c(1,N-1) \\ g_c(2,0) & g_c(2,1) & \cdots & g_c(2,N-1) \\ \vdots & \vdots & \ldots & \vdots \\ g_c(M-1,0) & g_c(M-1,1) & \cdots & g_c(M-1,N-1) \end{bmatrix}$$

where $g_c(i,j) = [g_R(i,j), g_G(i,j), g_B(i,j)]$ and $g_k(i,j) \in \{0,\ldots,L-1\}$ (for $k = R\ or\ G\ or\ B$) are non-negative integers. Usually $L = 255$.

Mathematical Morphology

Basics

- **Mathematical Morphology** is a nonlinear tool which is useful for the analysis and processing of geometrical structures in image.

Basics

- **Mathematical Morphology** is a nonlinear tool which is useful for the analysis and processing of geometrical structures in image.
  - It directly operates on shape.

Basics

- **Mathematical Morphology** is a nonlinear tool which is useful for the analysis and processing of geometrical structures in image.
    - It directly operates on shape.
    - It is defined using set theoretic operations.

Basics

- **Mathematical Morphology** is a nonlinear tool which is useful for the analysis and processing of geometrical structures in image.
  - It directly operates on shape.
  - It is defined using set theoretic operations.
  - It employs neighbourhood operators involving add (or subtract) followed by max (resp.min).

Elementary operations of Morphology

- **Dilation ($\oplus$):** The value of the output pixel is the maximum of sum of pixels values of the image and that of a kernel, i.e.,

$$(I \oplus S_d)(x, y) = \max_{i \in a; j \in b} \{I(x - i, y - j) + S_d(i, j)\}$$

  - **Structuring Element**: The kernel S, called *Structuring element* (SE), is defined over a small region or domain ($a \times b$).

Elementary operations of Morphology

- **Dilation ($\oplus$):** The value of the output pixel is the maximum of sum of pixels values of the image and that of a kernel, i.e.,

$$(I \oplus S_d)(x, y) = \max_{i \in a; j \in b} \{I(x - i, y - j) + S_d(i, j)\}$$

  - **Structuring Element**: The kernel S, called *Structuring element* (SE), is defined over a small region or domain ($a \times b$).

- **Erosion ($\ominus$):** The value of the output pixel is the minimum of subtracting kernel pixels from that of the image, i.e.,

$$(I \ominus S_e)(x, y) = \min_{i \in a; j \in b} \{I(x + i, y + j) - S_e(i, j)\}$$

Dilation and erosion: Examples

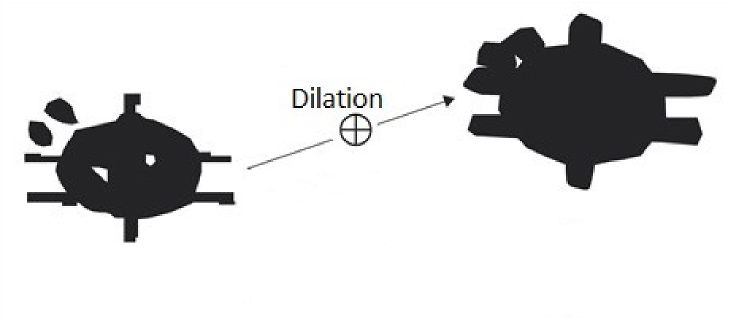


Figure: Results of Dilation and Erosion

Dilation and erosion: Examples

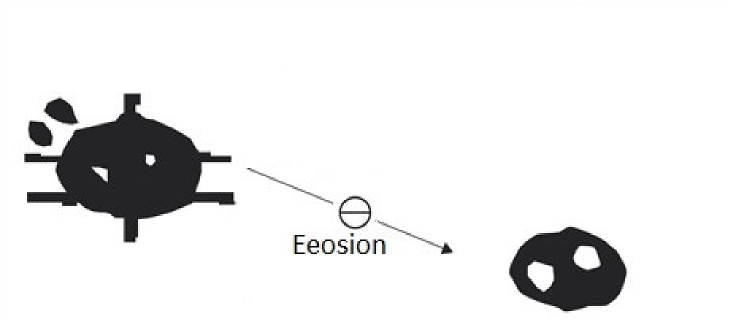


Figure: Results of Dilation and Erosion

Dilation and erosion: Examples

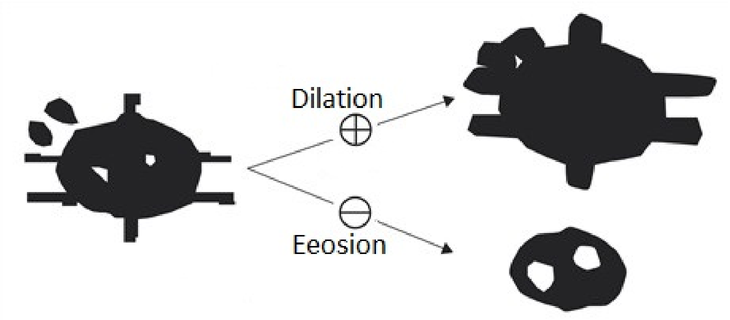


Figure: Results of Dilation and Erosion

Dilation with different SE

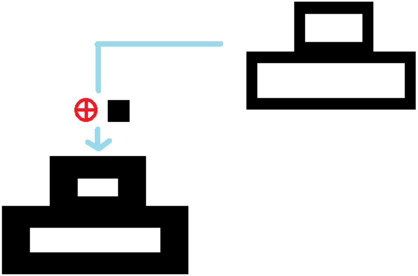


Figure: Results of Dilation.

Dilation with different SE

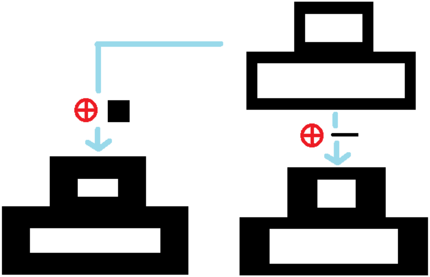


Figure: Results of Dilation.

Dilation with different SE

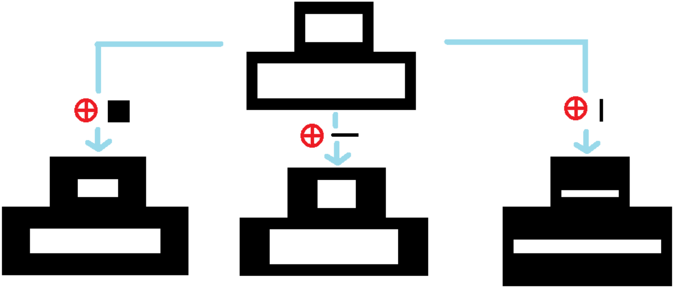


Figure: Results of Dilation.

Dilation Operation (gray-scale images)

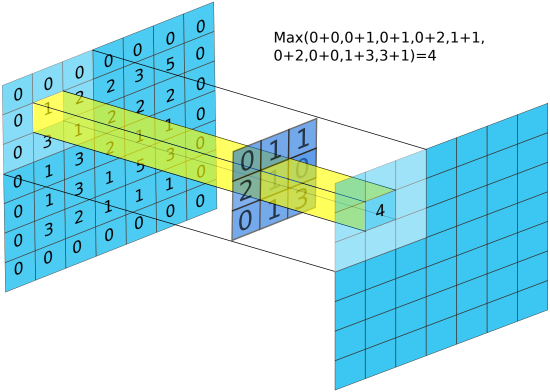


Figure: Dilation Operation with structuring elements of size 3×3

Noise cleaning: Example

Input: Gray scale image with noise
Output: Remove noise from the image

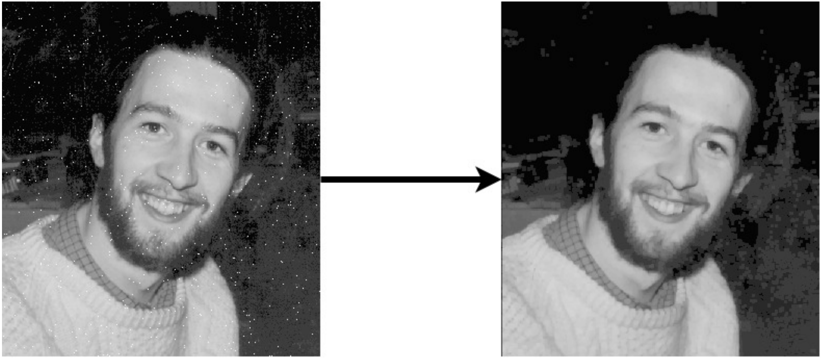


Figure: Result of erosion followed by a dilation with a 3×3 square SE.

Limitations

- Defining shape and size of structuring element suitable for the task.

Limitations

- Defining shape and size of structuring element suitable for the task.
- Determining operations to be used and their order.

Limitations

- Defining shape and size of structuring element suitable for the task.
- Determining operations to be used and their order.
- It is even harder when it comes to gray scale or color images.

Motivation

Can we learn the Structuring Elements and the sequence of operations?

ANN and CNN

Neuron and node

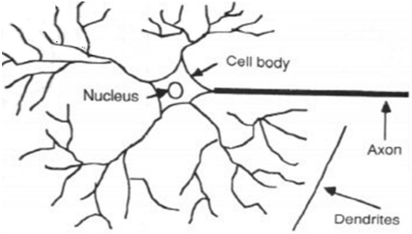


Figure: Biological neuron

Neuron and node

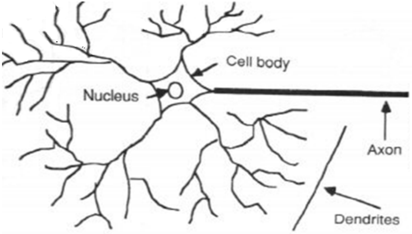


Figure: Biological neuron

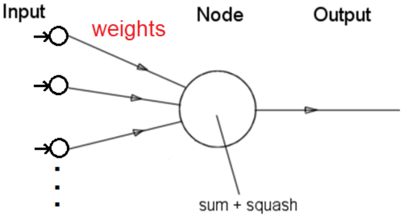


Figure: Artificial neural network node

ANN node

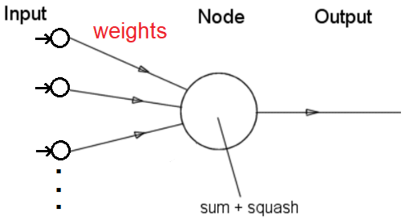


Figure: Artificial neural network node

At node: Output $y = \sigma(\sum(w_i f_i))$
where $\sigma(\cdot)$ is a *squashing / activation function.*

ANN node

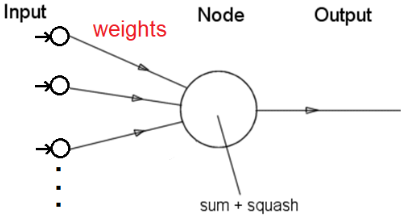


Figure: Artificial neural network node

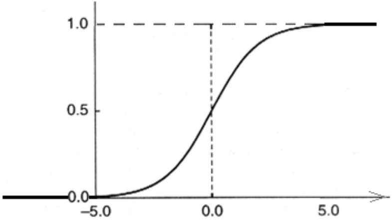
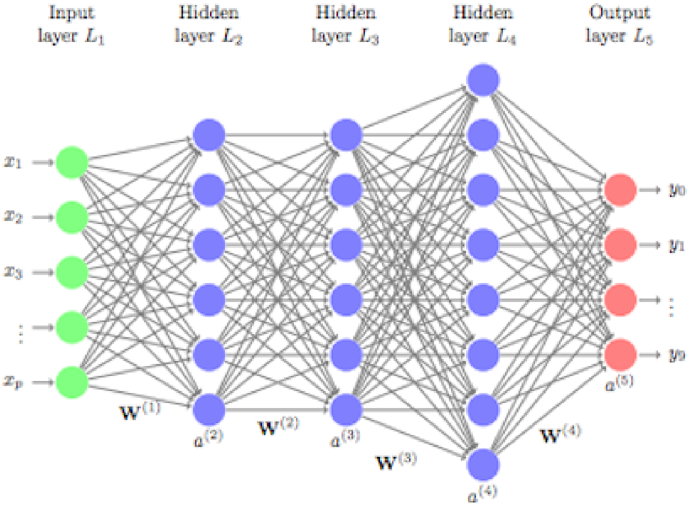


Figure: Squashing / activation function

At node: Output $y = \sigma(\sum(w_i f_i))$
where $\sigma(\cdot)$ is a *squashing / activation function.*

Multi-layer feedforward ANN



Convolutional neural network (CNN)

CNN has two parts: (i) feature learning (Conv, Relu and Pool) and (ii) Classification (Fully-connected network).

- CNN architectures for images:

Convolutional neural network (CNN)

CNN has two parts: (i) feature learning (Conv, Relu and Pool) and (ii) Classification (Fully-connected network).

- CNN architectures for images:
  - The explicit assumption that the inputs are images.

Convolutional neural network (CNN)

CNN has two parts: (i) feature learning (Conv, Relu and Pool) and (ii) Classification (Fully-connected network).

- CNN architectures for images:
  - The explicit assumption that the inputs are images.
  - Allows us to extract certain features.

Convolutional neural network (CNN)

CNN has two parts: (i) feature learning (Conv, Relu and Pool) and (ii) Classification (Fully-connected network).

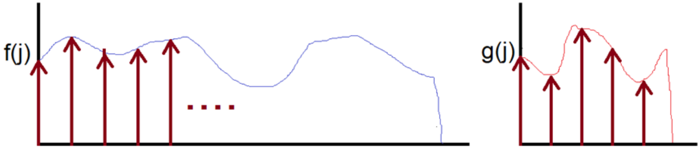
- CNN architectures for images:
    - The explicit assumption that the inputs are images.
    - Allows us to extract certain features.
    - Large images do not fit into fully-connected structure.

Convolutional neural network (CNN)

CNN has two parts: (i) feature learning (Conv, Relu and Pool) and (ii) Classification (Fully-connected network).

- CNN architectures for images:
    - The explicit assumption that the inputs are images.
    - Allows us to extract certain features.
    - Large images do not fit into fully-connected structure.
    - Pooling vastly reduces the number of parameters.

1D Discrete function

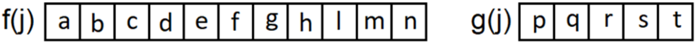


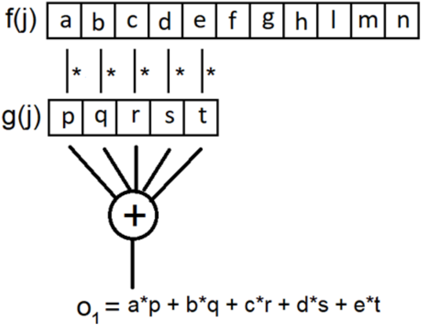
f(j)	a	b	c	d	e	f	g	h	l	m	n

g(j)	p	q	r	s	t

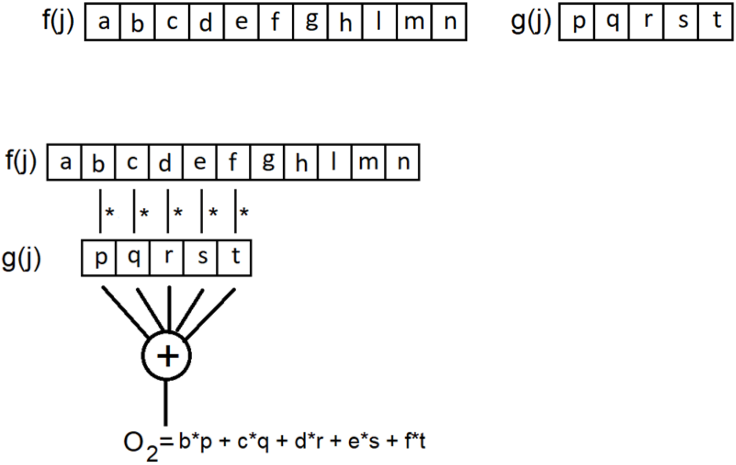
$$\text{1D Convolution}: \quad (f * g)(k) = \sum_j g(k-j) f(j)$$

1D Discrete Convolution

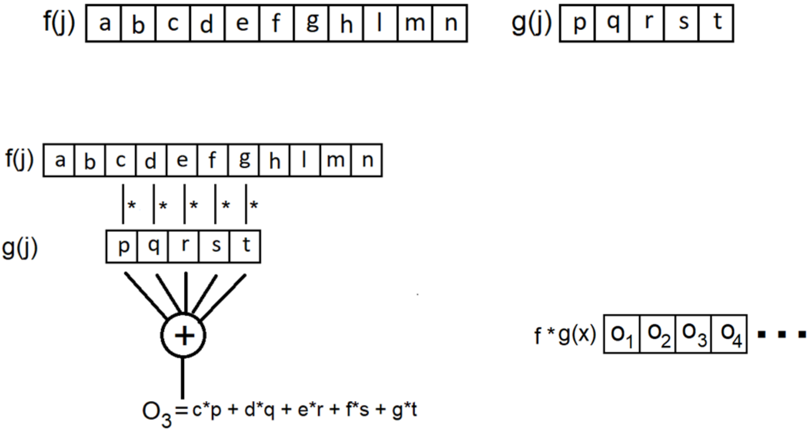




1D Discrete Convolution



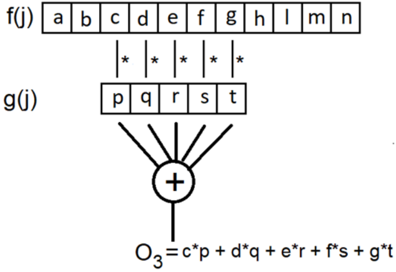
1D Discrete Convolution

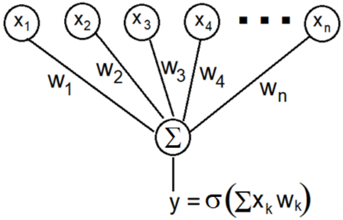


1D Discrete Convolution

$$O_k = (f * g)(k) = \sum_j g(k-j)f(j)$$

$$y = \sigma(\sum_j w_j x_j)$$





Convolutional neural network (CNN)

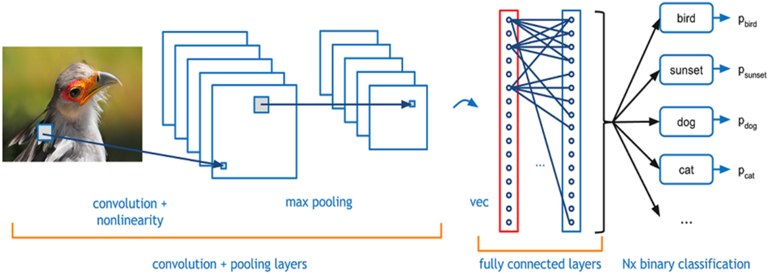


Figure: Illustrates a basic CNN for classification[1].

Convolution kernel along with other parameters is learned based on feedback due to error between output of the network and the ground truth.

[1] Liu et al., A survey of deep neural network architectures and their applications. Neurocomputing 234: 11-26 (2017)

Morphological and Convolution Operators: Similarity

Both are neighbourhood operators.

- **Convolution($*$):** The value of the output pixel is the sum of product of kernel pixels and that of the image, i.e., $(I * S_c)(x, y) = \sum_{i \in a; j \in b} (I(x + i, y + j) S_c(i, j))$

Morphological and Convolution Operators: Similarity

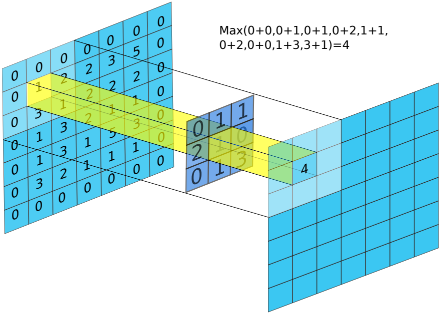
Both are neighbourhood operators.

- **Convolution($*$):** The value of the output pixel is the sum of product of kernel pixels and that of the image, i.e.,
$(I * S_c)(x, y) = \sum_{i \in a; j \in b} (I(x + i, y + j) S_c(i, j))$

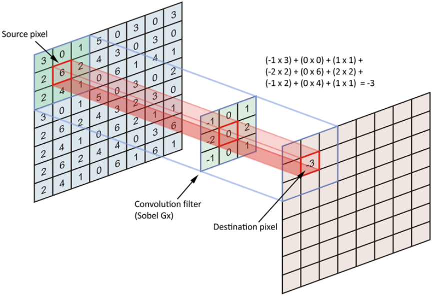
- **Dilation($\oplus$):** The value of the output pixel is the maximum of sum of pixels values of the image and that of a kernel, i.e.,
$(I \oplus S_d)(x, y) = \max_{i \in a; j \in b} \{I(x - i, y - j) + S_d(i, j)\}$

- **Erosion($\ominus$):** The value of the output pixel is the minimum of subtracting kernel pixels from that of the image, i.e.,
$(I \ominus S_e)(x, y) = \min_{i \in a; j \in b} \{I(x + i, y + j) - S_e(i, j)\}$

Morphological and Convolution Operations: Similarity



(a) Dilation



(b) Convolution

Figure: Comparison with Convolution and Dilation operation

Automatic Learning of Structuring Element

De-raining problem

Input: Gray Scale Rainy Image
Expected **Output**: Gray Scale De-rained Image

Figure: Left: Rainy image ; Right: De-rained desired Image

Challenges

- In reality different rainy images have rain drops of different shapes and sizes.

Challenges

- In reality different rainy images have rain drops of different shapes and sizes.
- What would be the shape and size of the structuring elements?

Challenges

- In reality different rainy images have rain drops of different shapes and sizes.
- What would be the shape and size of the structuring elements?
- Finally, in which order morphological operators be applied to solve the problem?

Learning Morphological Network

- Build a network using dilation and erosion operation.

Learning Morphological Network

- Build a network using dilation and erosion operation.
- Randomly initialize all the structuring elements of the network and define a loss function.

Learning Morphological Network

- Build a network using dilation and erosion operation.
- Randomly initialize all the structuring elements of the network and define a loss function.
- Learn morphological structuring elements with back-propagation algorithm.

Learning Morphological Network

- Build a network using dilation and erosion operation.
- Randomly initialize all the structuring elements of the network and define a loss function.
- Learn morphological structuring elements with back-propagation algorithm.
  - Dilation (Erosion) involves *max* (resp. *min*) operation.
  - *max* and *min* are not fully differentiable. However, they are piecewise differentiable.

Network can be presented pictorially as

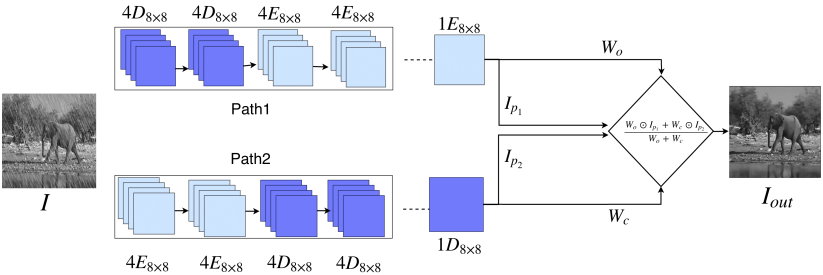


Figure: Morphological Network for Image De-raining

Network can be presented pictorially as

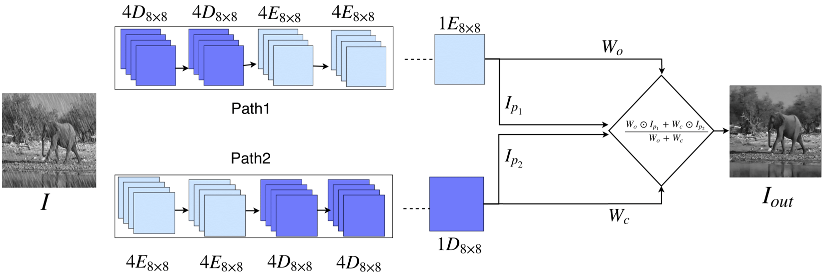


Figure: Morphological Network for Image De-raining

- Path 1 works like a kind of removing dark noise
- Path 2 works like a kind of removing white noise
- Multiple dilation and erosion layers can deal with varied sizes of artifacts.

Qualitative Results

(a) Input (b) Path1 (c) Path2 (d) Morph-Net

Quantitative Results

Table: Comparison of Morpho-Net[2] with a CNN (U-Net)[3] on rain dataset[4].

Metric	Input	CNN	Path1	Path2	Morph-Net
#Parameters	-	6,110,773	7,680	7,680	16,780
#Params w.r.t. CNN	-	100.0%	0.12%	0.12%	0.27%
SSIM	0.85	**0.92**	0.87	0.90	**0.92**
PSNR	24.3	**29.12**	26.27	27.20	28.03

[2]Mondal et al., Morphological networks for image de-raining. Int. Conf. DGCI, 262-275, Springer, 2019.

[3]Ronneberger et al., U-net: Convolutional networks for biomedical image segmentation, ICMICCI, Springer, 234–241, 2015.

[4]Fu et al., Clearing the skies: A deep network architecture for single-image rain removal, IEEE TIP 26.6, 2944–2956, 2017.

Observations

- Structuring Elements can be learned in form of network.
- It needs less number of parameters compared to CNN.

Opening-Closing Network

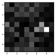
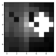
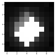
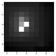
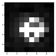
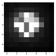
Opening and Closing Operations

$$\text{Opening:} \quad (X \circ S)(x,y) = ((X \ominus S) \oplus S)(x,y) \tag{1}$$

$$\text{Closing:} \quad (X \bullet S)(x,y) = ((X \oplus S) \ominus S)(x,y) \tag{2}$$

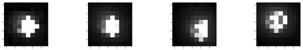
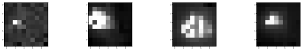
Simulation of opening and closing

Table: Network architecture in 2nd column; the operation suggested in 1st column

Morphological Operation to be Achieved	Network Architecture	Learned Structuring elements from the first layer
Dilation	$X \to E^1_{10\times 10} \to D^1_{10\times 10} \to Y$	 
Erosion	$X \to E^1_{10\times 10} \to D^1_{10\times 10} \to Y$	 
Opening	$X \to E^1_{10\times 10} \to D^1_{10\times 10} \to Y$	 

Simulation of opening and closing

Table: Network architecture in 2nd column; the operation suggested in 1st column

Morphological Operation to be Achieved	Network Architecture	Learned Structuring elements from the first layer
Opening	$X \rightarrow E^1_{10\times10} \rightarrow D^1_{10\times10} \rightarrow E^1_{10\times10} \rightarrow D^1_{10\times10} \rightarrow Y$	
Closing	$X \rightarrow E^1_{10\times10} \rightarrow D^1_{10\times10} \rightarrow E^1_{10\times10} \rightarrow D^1_{10\times10} \rightarrow Y$	

Dilation: examples

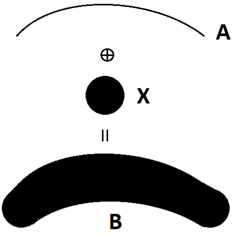


Figure: Dilation by different structuring elements.

Dilation: examples

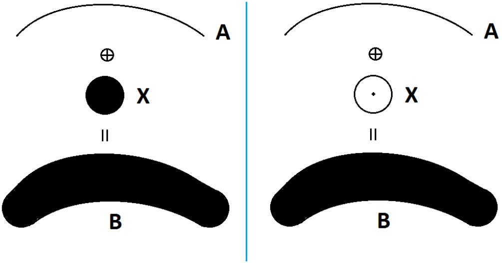


Figure: Dilation by different structuring elements.

Dilation: examples

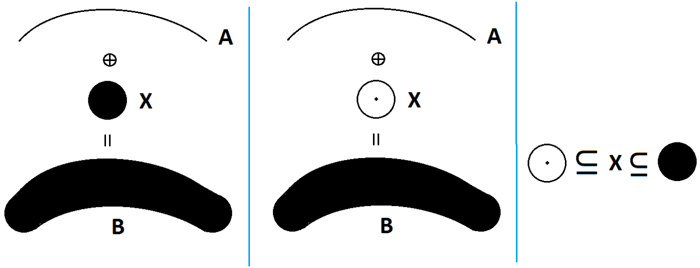


Figure: Dilation by different structuring elements.

Opening-Closing network for image de-raining

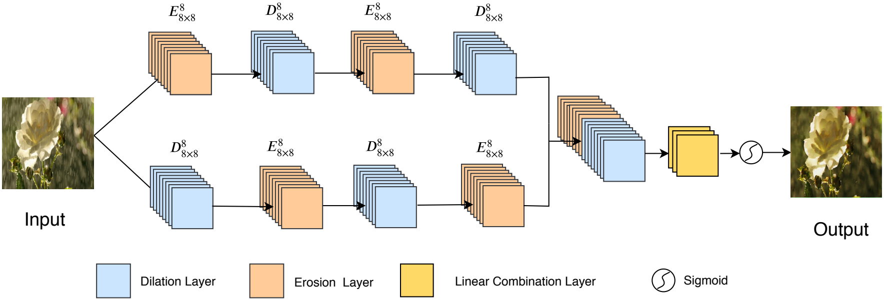


Figure: The network for color image de-raining. It consists of 2 parallel paths containing a complementary sequence of Erosion and Dilation layers.

Qualitative Results on Image de-raining

(a) Input Frame (b) Opening-Closing Net (c) CNN (d) Ground Truth

Figure: Result of De-rain network over color images from Rain dataset. In few places of output from CNN gets blurred compared to opening-closing net.

Quantitative Results on Image de-raining

Table: Results of de-raining on both grayscale and color images of Rain dataset.

	Metric	Base Line	Closing Net	Opening Net	Opening-Closing Net	CNN
Gray	SSIM	0.85	0.90	0.90	0.91	**0.93**
	PSNR	24.41	26.00	25.99	27.29	**29.24**
Color	SSIM	0.84	0.88	0.87	0.89	**0.91**
	PSNR	24.06	24.81	24.28	25.01	**27.33**

Image De-hazing using Opening-Closing Network

De-hazing Problem

(a) Image without haze (b) Hazy image

Figure: Effect of haze on image.

Opening-Closing network for image de-hazing

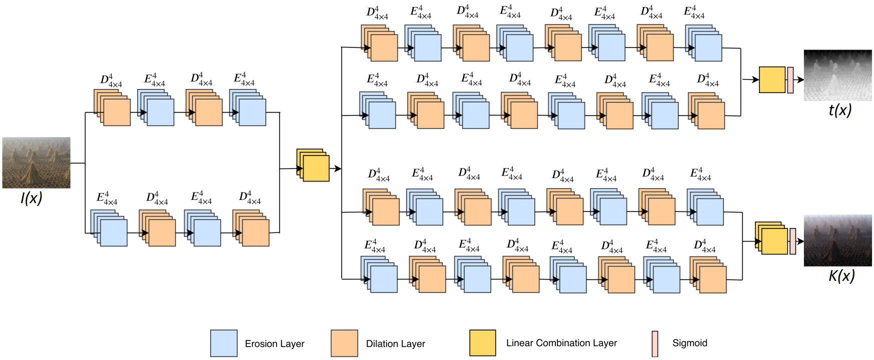


Figure: The network for image de-hazing. It consists of 2 parallel paths containing a complementary sequence of Erosion and Dilation layers.

Image de-hazing by Morpho-net on O-HAZE dataset

(a) Input Image (b) Transmittance (c) Airlight (d) Output (e) Ground Truth

Figure: Results of opening-closing de-haze network over O-HAZE dataset. The transmittance and airlight map along with ground truth is shown for qualitative evaluation.

Conclusion

- Morphological Network may be used parallel to Neural Network or CNN.
- Use of Max/Min operation increases the training time.
    - Training time may be reduced by adopting soft dilation and erosion.
- Morphological operations need fewer parameters to be trained compared to NN or CNN.
- The use of sum-Max (or diff-Min) operation in a morphological network is computationally efficient than use of multiplication-sum-squash in NN or CNN.

Thank You